

Booster damper logbook

progress notes by Bill Ashmanskas

2008-06-30

Decided to work with old Foster/Semenov board for now, as I can get up to speed immediately and return to where JP and I were in 2006.

Meanwhile, I have asked Sten whether he might be able to get Alexey's DDR interface working in the new Eddy/Larwill board.

First step: input timing. I took the vertical difference signal and fed it into ADC channel 1. There used to be an amplifier in between, so I expect to have to add cable delay. Miraculously, the FIFO readout worked on the first try.

Graphing $\text{atan2}(q,i)$ vs. RF frequency was a handy trick. Saw 100 degrees / 10 MHz \Rightarrow 28 ns. Added 32 ns of cable, now see -12.6 degrees / 10 MHz \Rightarrow -3.5 ns. Makes sense! Tweaked cable a bit more, got phase vs. frequency flat to ~ 1 ns of cable. Now nearly all pulse height is in the Q samples, not I.

ADC values reach ~ 1200 counts at end of cycle, which is about half of full-scale. 12-bit ADCs have $\pm 2K$ range. This was after replacing 20dB pads with 10dB pads before difference hybrid.

Next, worked on following what's happening down through the pipeline. Did Fast Time Plot of B:CHG0 (about $5E12$), D4YQDC[bucket], D4YQDS, which are bunch and summed (for a whole turn) quadrature values. Bill F. doubles the sum of 2 samples for "quadrature," which gives $\pm 8K$ range. ACNET FTP I:D4YQDC peaks at just under 5000 counts at end of ramp, which is consistent with FIFO readout. I:D4YQDS reaches about 375K, which is close enough to 84×5000 .

OK, now I see that I want to look through DdY:damper_channel_logic and DdC:damper_control_logic for the next stages (after IQ_logic, etc.).

damper_channel_logic takes quadrature[15..0], damper_control_enable and returns kick[], filter[]

damper_control_logic ...

I'm also thinking more about the FIFO. It looks to me as if I may be able to make the FIFO hold about 7500 full turns' data, instead of just over 600. Each FIFO holds about 212K 20-bit words and is clocked at $4 \times RF$. Clocking at $1 \times RF$ gives a factor of 4. Using 3 FIFOs in parallel to hold Quadrature data for just channel #1 gives another factor of 3. That would be sufficient for working with this board indefinitely.

I started cleaning things up in damper.tdf, and now I seem to be in big trouble. I messed up the "greeting" message from the Stratix to the SHARC, so port 128 doesn't work anymore. I tried to back out by modifying rewrite_stratix.py to use port 100, and that almost worked, but not quite. I think I need to reload the stratix manually. Let's hope I don't need to reload the SHARC manually! Arrgh!

By the end of the day, I managed to use my Mac to reload the Stratix by hand, which then allowed me to run the python program to load the Stratix code into the FRAM. Phew!

2008-07-01

My big goal for today is to get the FIFO capacity from ~ 600 turns to ~ 7000 turns (for all 84 bunches) by using three FIFOs round-robin at $1 \times \text{RF}$. That gains me a factor of 12. Thus far (2pm), I have succeeded in writing the bunch/turn counters to the FIFOs as desired. Now I'm sticking the Quadrature[] signal in there.

Thoughts on dividing the work:

- Bill - old board, FIFO, demonstrate 1 plane ASAP (\$17 cycles)
- Sten - make Alexey's DDR code work on Nathan's board
- Meghan - use scope/damper to observe beam cycles, analyze with scipy
- ??? - port everything to Nathan's board

I see that the FIFO depth is in fact 256K (I don't know why I thought 212K -- makes little sense), so I can do 9300 turns! FIFO part number is IDT72T20128_2.

I am now getting good stuff out of the FIFO, for 9300 turns, basically from transition to end of cycle!

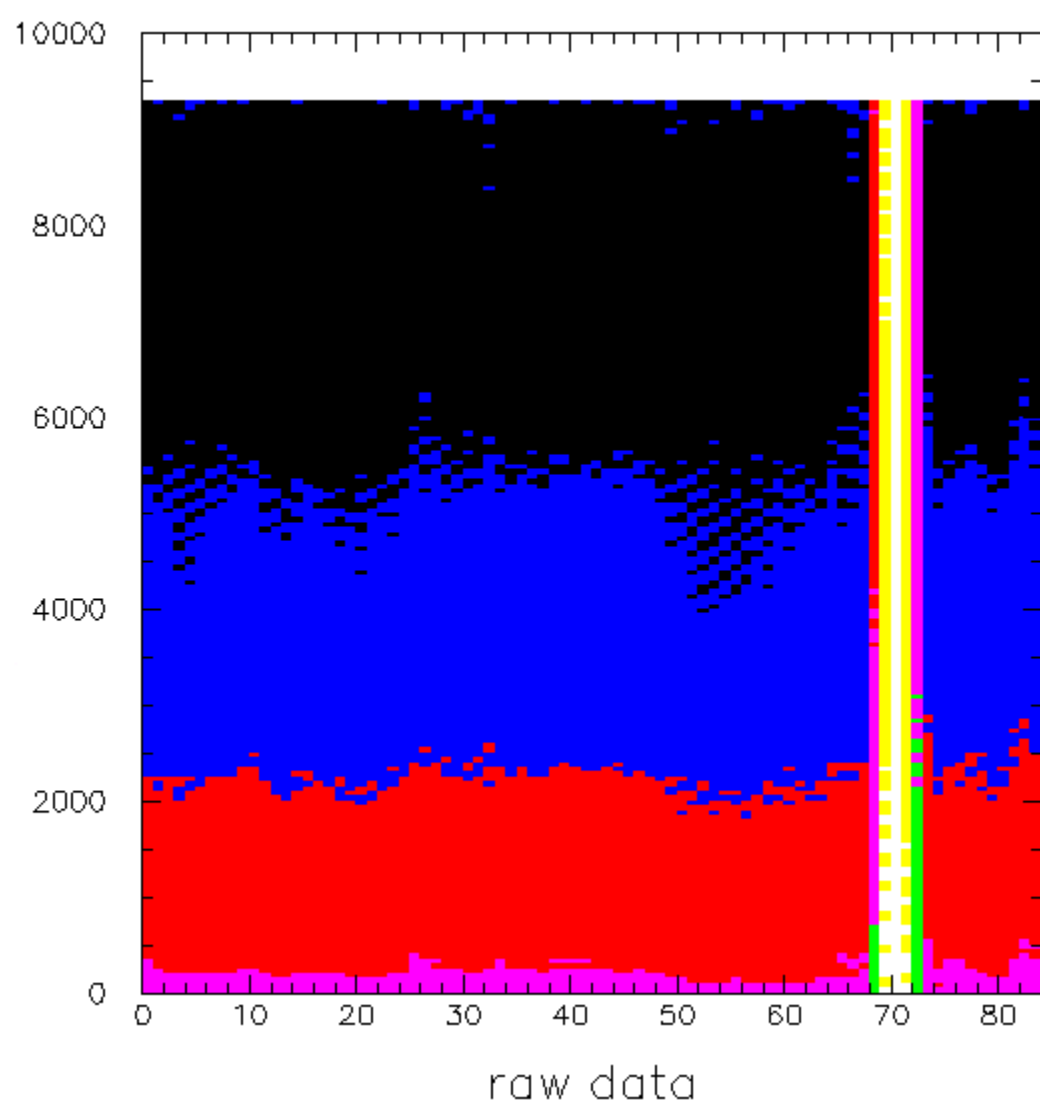
Quality of life issues (for rapid progress):

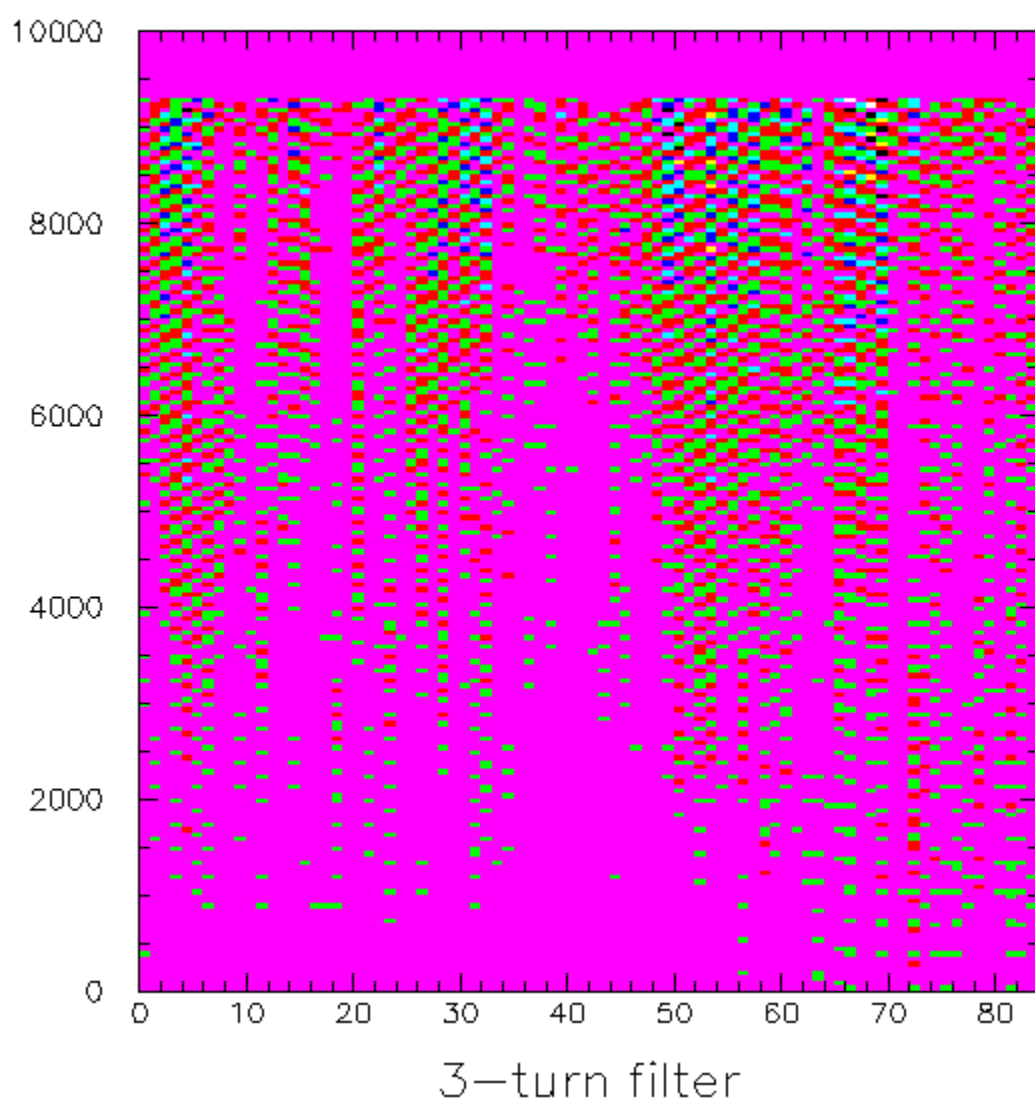
- Need to get a decent (laptop?) computer over by the damper
- Altera compile time is now 9 minutes -- not bad
- FIFO readout time is 6 minutes (7 KB/s!!) -- could be much better
- Figure out how to do software beam switch for Booster

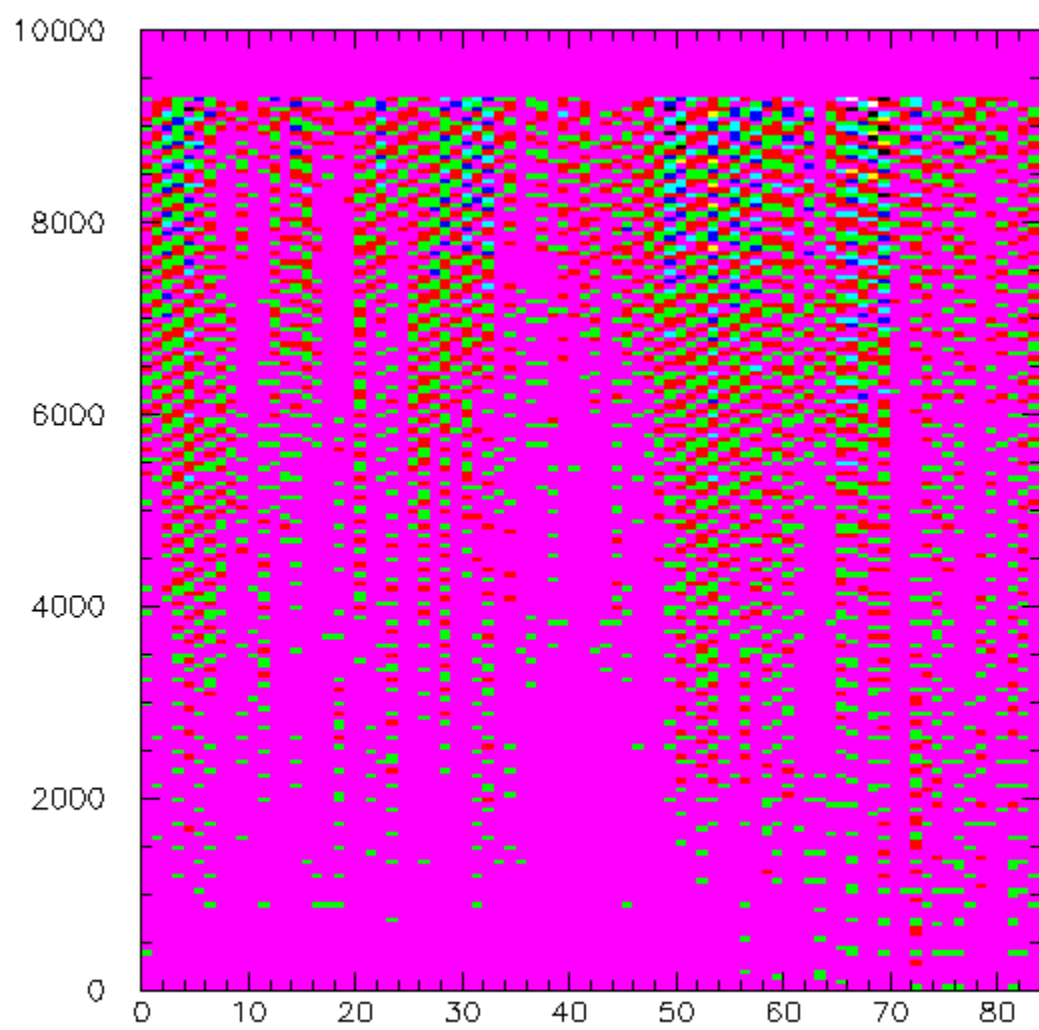
Next steps (on real progress, perhaps distracted part-time by QOL improvements):

- Use scipy to filter FIFO data and observe betatron oscillations
- Cook up a nice way to graph coherent betatron signal vs. bucket & turn
- Get Bill Pellico to adjust the longitudinal damper gain so that we can see the effect
- Look at damper kick waveform on scope and make sure it has right shape
- Look at timing of damper kick waveform on scope, to get pipeline delays right
- Get some \$17 cycles and verify kicker timing
- Repeat earlier antidamping/damping experiments on \$17 cycles

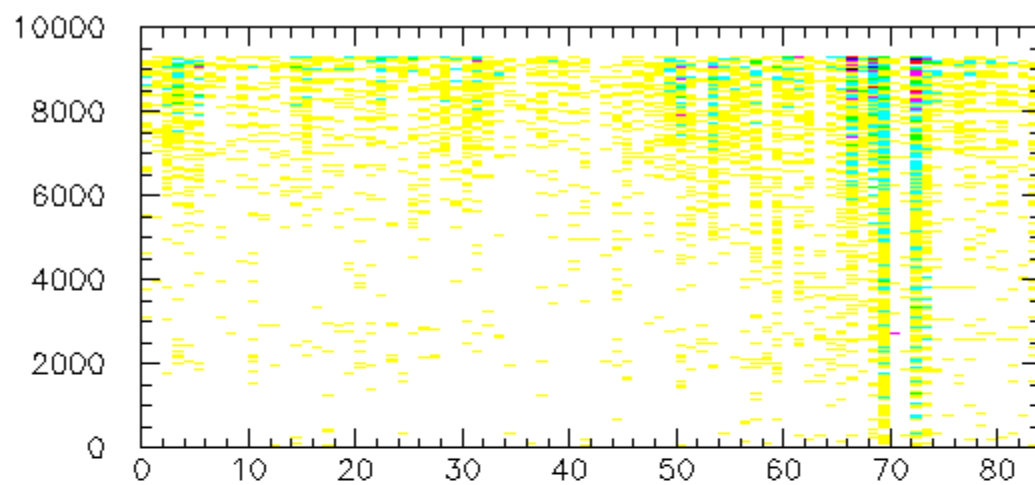
OK, progress! I got python/scipy to read the data file that I write by reading out the damper's FIFOs. Then I compute the two-turn difference (turn n minus turn $n-1$) and the three-turn difference (which I take for now to be turn n minus turn $n-2$). Since the fractional tune is not so far from 0.75 (it should be something in the 0.7 to 0.8 range -- which I could just extract from these data), $x(n)$ minus $x(n-2)$ should be pretty decent. The "turn" axis here is really "turn minus 11300."



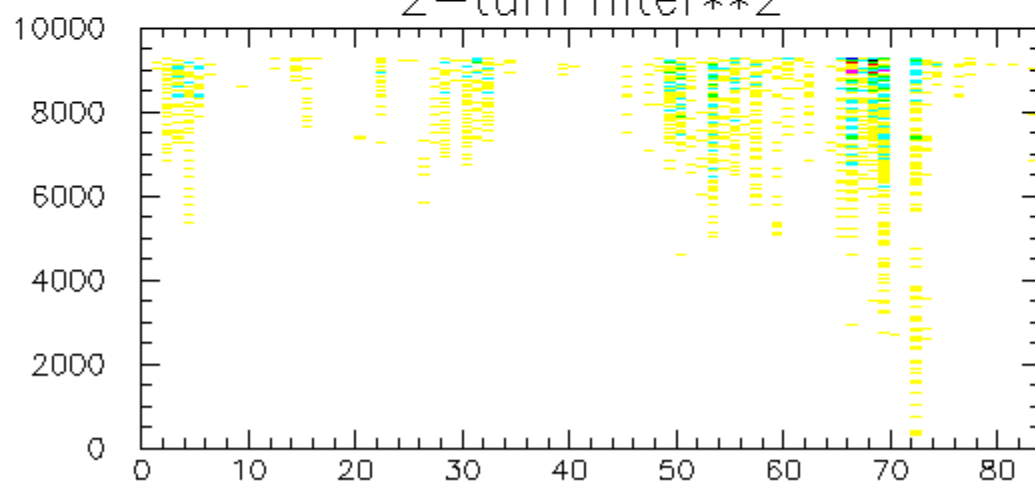




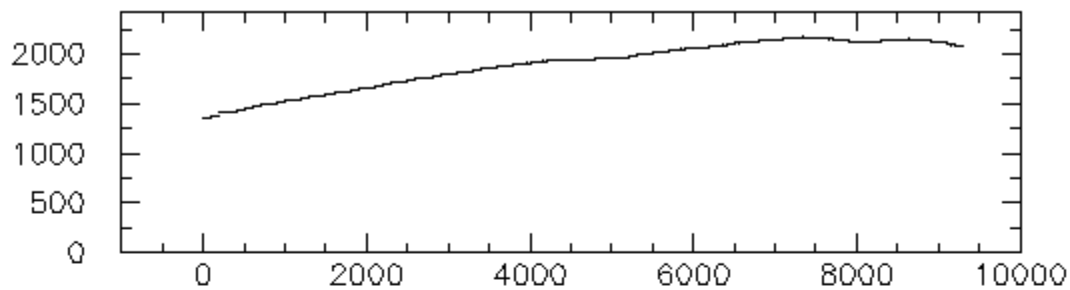
2-turn filter



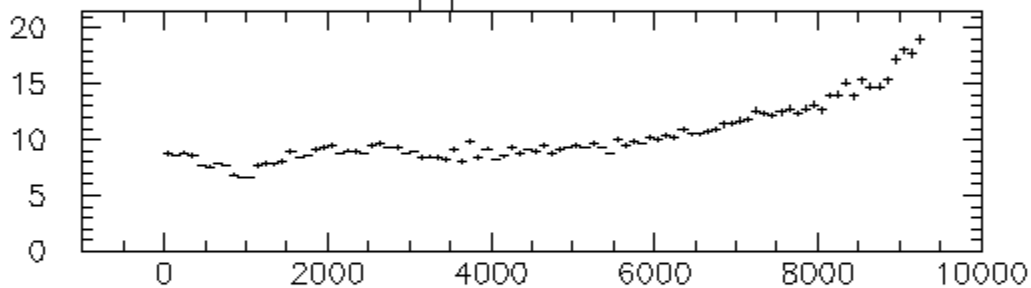
2-turn filter**2



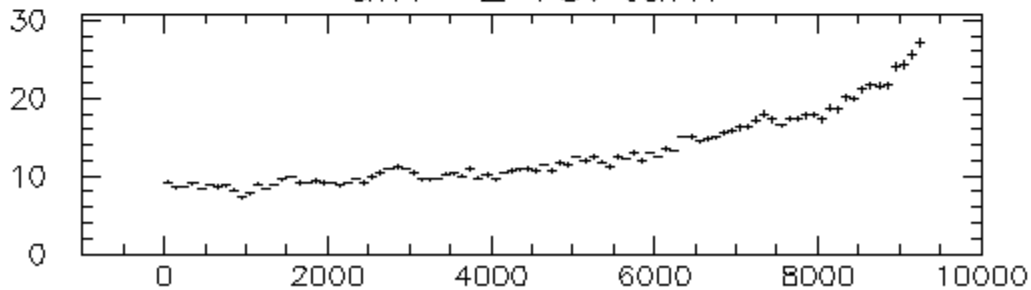
3-turn filter**2



qq VS. turn



diff**2 VS. turn



(filt/7)**2 VS. turn

2008-07-02

I found a C++ example `tcp-client.cc` on the web, and modified it to read out the damper FIFOs. I found that it was only about 10-20% faster than my python program, when using the (hexadecimal) `FIFOREAD` command that I had implemented on the SHARC back in 2004. Then I looked through the code and saw that I had, in fact, implemented a binary fifo read command (FRB). It is not fully efficient in network bandwidth (4 bytes per FIFO word vs. 6 (space + 5 hex digits), when it could send 3 bytes per 20-bit word), but it avoids the hexadecimal conversion. Remarkably, it lets me read out one FIFO (of four) in 11 seconds, using the C++ program, running on `balrog.fnal.gov` (over in AP50, on the protected subnet). Then I tried doing it again in python, since the code is so much more flexible that way. The timing is 11.105 seconds in C++, 11.115 seconds in python, even on a 666 MHz Pentium III. So I'll stick with python.

OK, now the program that sets up the board, waits for the trigger, and empties the FIFOs takes just 55 seconds. Much better than 6 minutes!

Now that I'm processing binary data, I need to worry about byte order. My SHARC code writes integers out byte-by-byte, MSB first (which follows the Internet byte-order convention). OK -- got it. So one simple python program now sets up the damper board, triggers the FIFO readout, waits for event, reads out the FIFOs, chops up the data, and writes a binary file that can be read by another program, e.g. to make graphs.

I showed Bill Pellico the data collection. He increased the longitudinal damper gain above transition on minibooone cycles once or twice, but the monitoring in the MI8 line tripped off the beam before we were able to record data for a pulse. Losses in Booster were not a problem, just in the MI8 line. So we'll repeat this tomorrow using \$17 cycles.

I have two digital oscilloscopes out by the damper. `pbar-scope-03` is a TDS3014. `jorkins` is a Tek 7000 series scope with high bandwidth and deep memory. The advantage of the former is that it has a web-based display of the scope screen, so it is good for quick tuning, while the second is better for reading out lots of trace data.

I have the 3014 scope triggering (via:B:HFDOFF) on \$1D (was \$14, oops), which is about 1.94 ms before beam arrives. Beam lasts about 33.5 ms. Scope ch1 = horizontal sum signal, ch2 = damper CH1 DAC (kick signal), ch3 = damper_active TTL output, ch4 = rev_marker_out TTL damper output.

Booster TCLK events of interest: \$10 = reset, \$14 = pbar, \$17 = study, \$19 = NuMI, \$1D = minibooone.

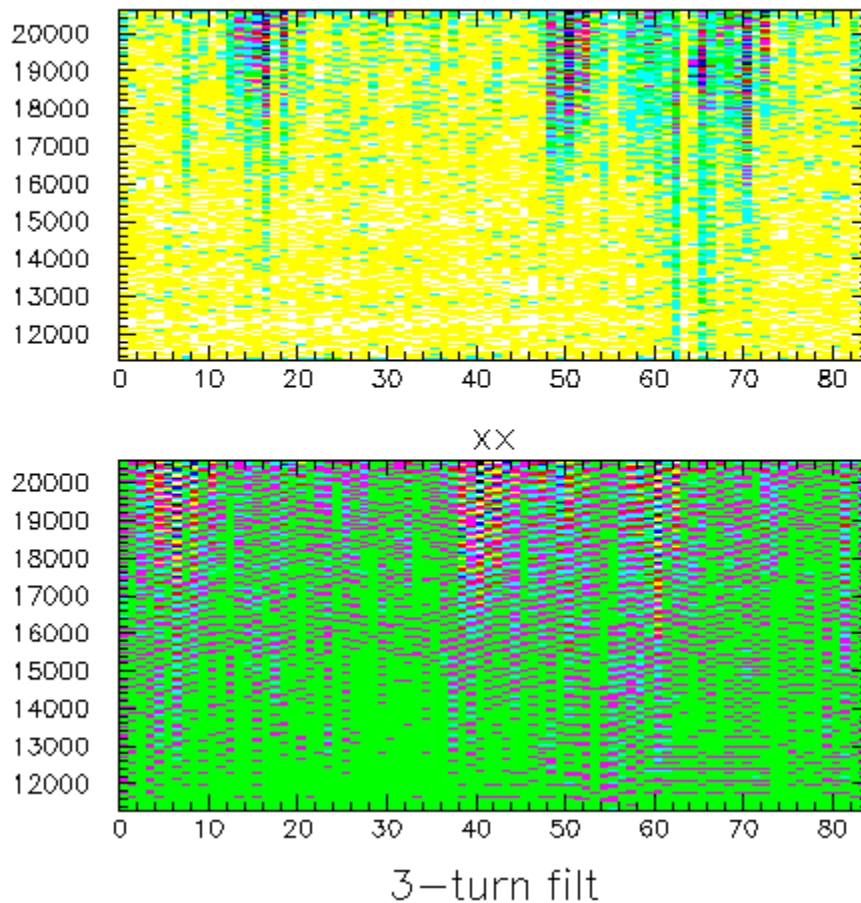
I have `rewrite_stratix.py` updating firmware in about half the previous time (2 minutes down to 1 minute). It was wasting lots of time reading the .tff data in an inefficient manner. I am reducing lots of complication from the damper firmware, so that I can implement something simple and quickly tune it up.

I removed from the code base pretty much everything that does not look relevant to this project. This is a step toward eventually making the design dramatically simpler overall.

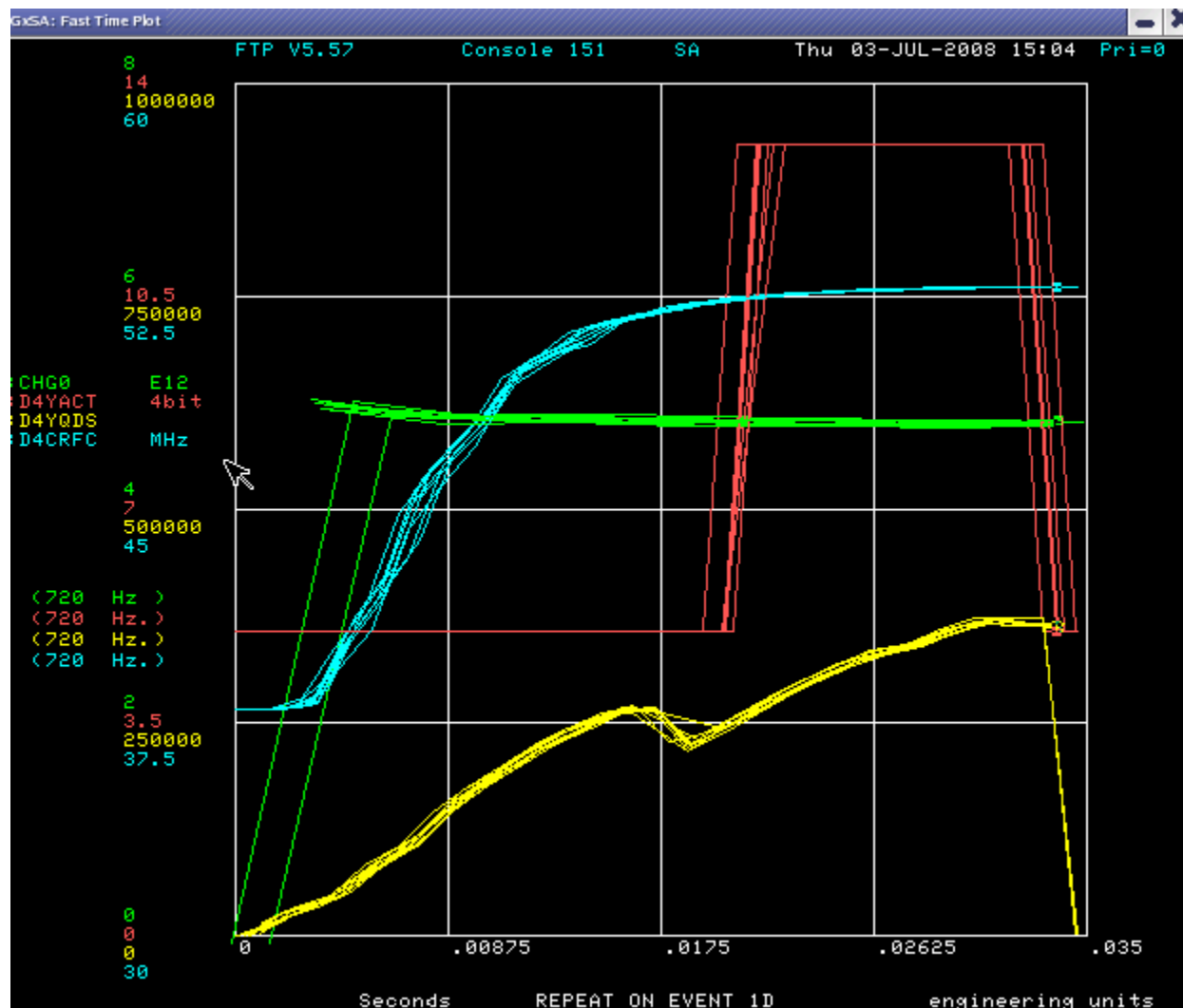
I have so far not seen any output from the DAC, whereas clearly the DAC was producing output two years ago. I first looked for signs of life from the DAC yesterday afternoon. I am tracing signals through the various stages of logic. I see that on the \$1D cycle, above transition, `I:D4YACT==13`, indicating that the damper is enabled and intending to damp. I also see that the quadrature signal looks good, of course (also verifiable by FTP of `I:D4YQDS`).

Now I want to look at the 3-turn filter, or various intermediate stages thereof, to see whether the results are nonzero. Phil and I were discussing at lunch yesterday the fact that it would be a very handy diagnostic to have something equivalent to `I:D4YQDS` for many other pipeline stages, i.e. some useful intermediate value, presumably with its absolute value summed over bunches and latched at the end of the turn, and made available for fast-time plot. Then when the damper is not doing anything sensible, one can simply make fast-time plots of the various intermediate stages and see what is happening. I will definitely add that, as if it already existed, I would be making faster progress right now. For now, I have added 7 bits of 3-turn filter output to the FIFO data for each bucket, am recompiling, etc. 12 minutes after editing the `damper.tdf` source file, I see filter values in

the FIFO that look pretty similar to the filter values computed by subtracting $q[n]-q[n-2]$. They are displaced several clock cycles w.r.t. the quadrature data, because I didn't make any attempt to line them up with different pipeline delays for the FIFO.

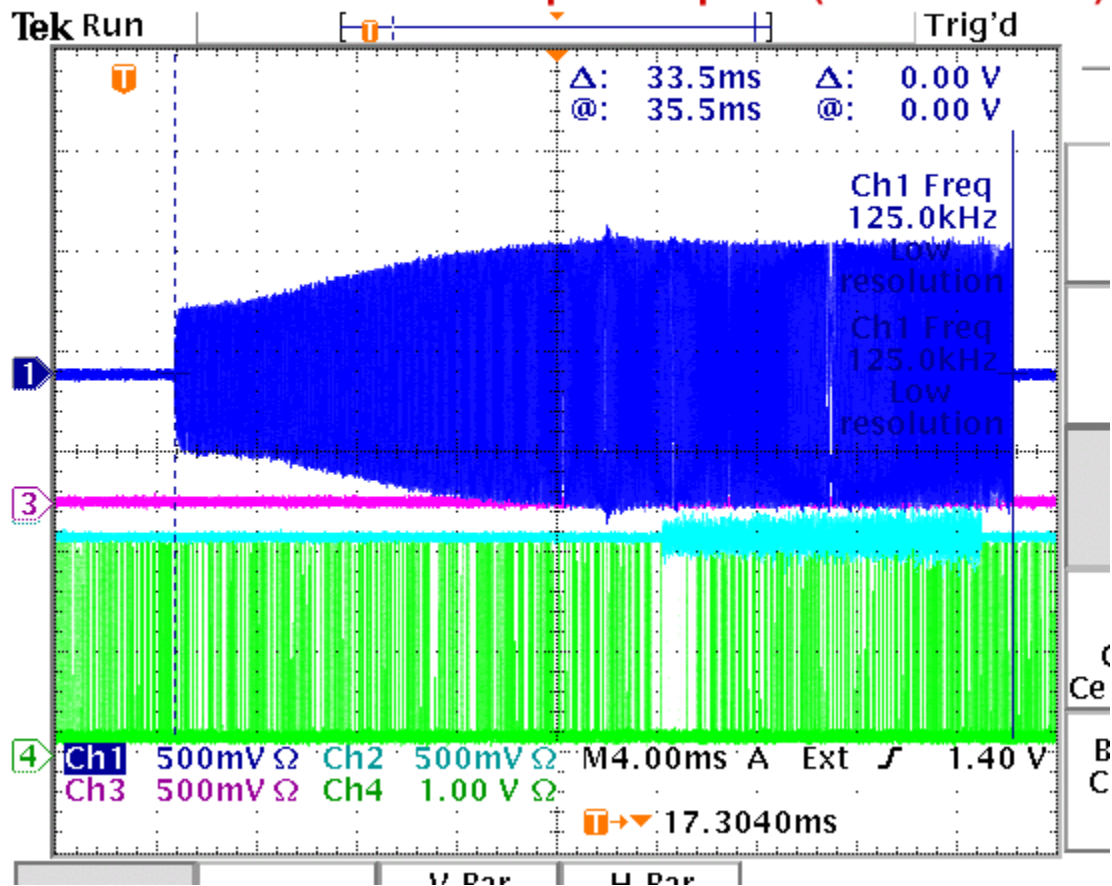


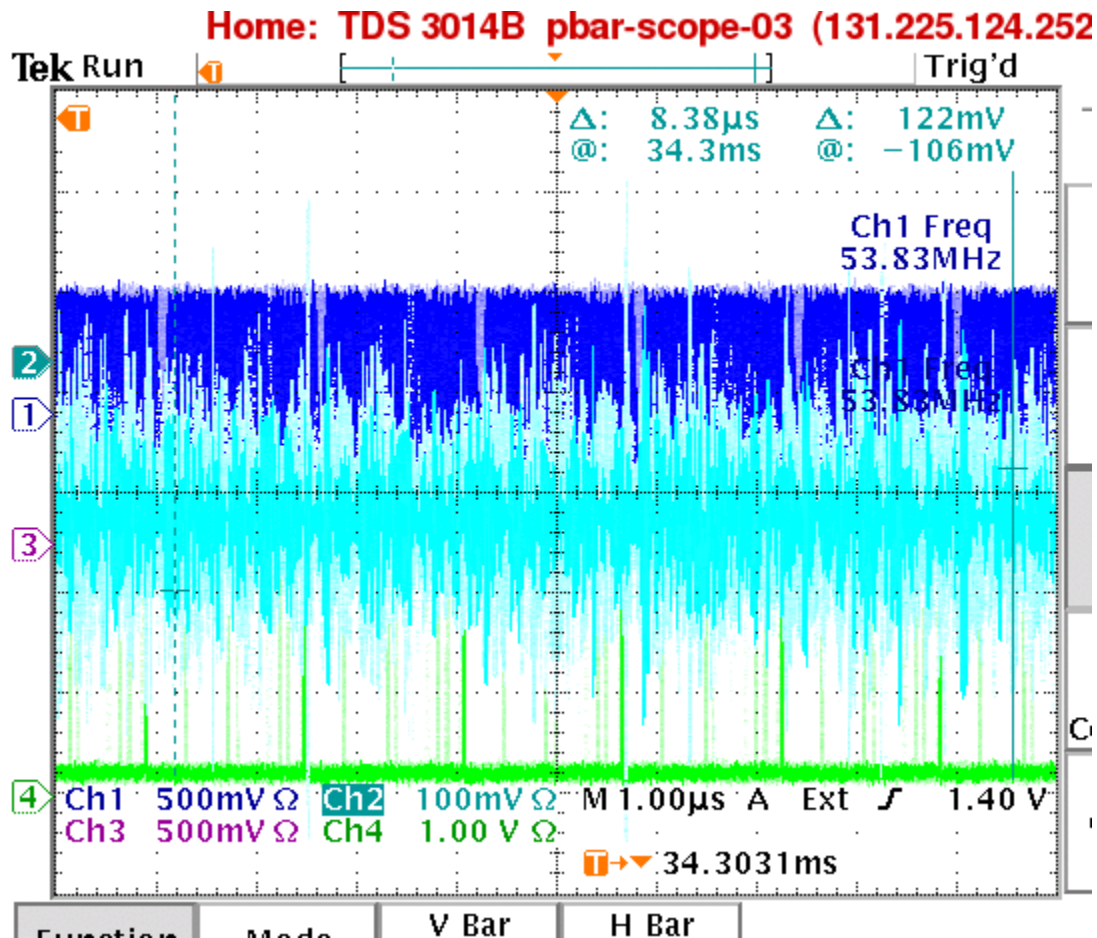
By the way, here is a fast time plot from the Booster damper, etc., showing B:CHG0 (booster intensity), I:D4CRFC (the damper board's measurement of the RF frequency), I:D4YQDS (the "quadrature" signal for the vertical difference cable, summed over bunches for a turn at a time), and I:D4YACT (indicating the damper's activation status: 13 = damping active). Clearly, it would be nice to be able to graph several intermediate results in this way, as a diagnostic. That sort of feature is also important for seeing whether intermediate stages are saturating, etc.



Hey, what do you know. After turning the gain all the way up in the damper's internal logic, I actually see a kick signal coming out of the board! The kicker (DAC) signal is the cyan waveform. As instructed, it is active from about 1000 turns after transition to about 500 turns before extraction.

Home: TDS 3014B pbar-scope-03 (131.225.124.252)





This would be better with persistence turned off. You can see the notch every 1.6s in the beam signal. I think I could benefit from higher gain in D4YBBG or negative bit shifting in D4YGAI. Also, I should tee past the TDS3014 scope and put the signals into the Tek 7000 series scope for readout and detailed analysis.

I wonder if a 7-turn filter would be cleaner ... A quick look at data offline says that it would not be particularly cleaner. I think part of the problem here is that there is not really much of a coherent oscillation to see. I'll see how clean the signals look when I have deliberately excited several bunches.

2008-07-07

Here are some things recently checked off of to-do list:

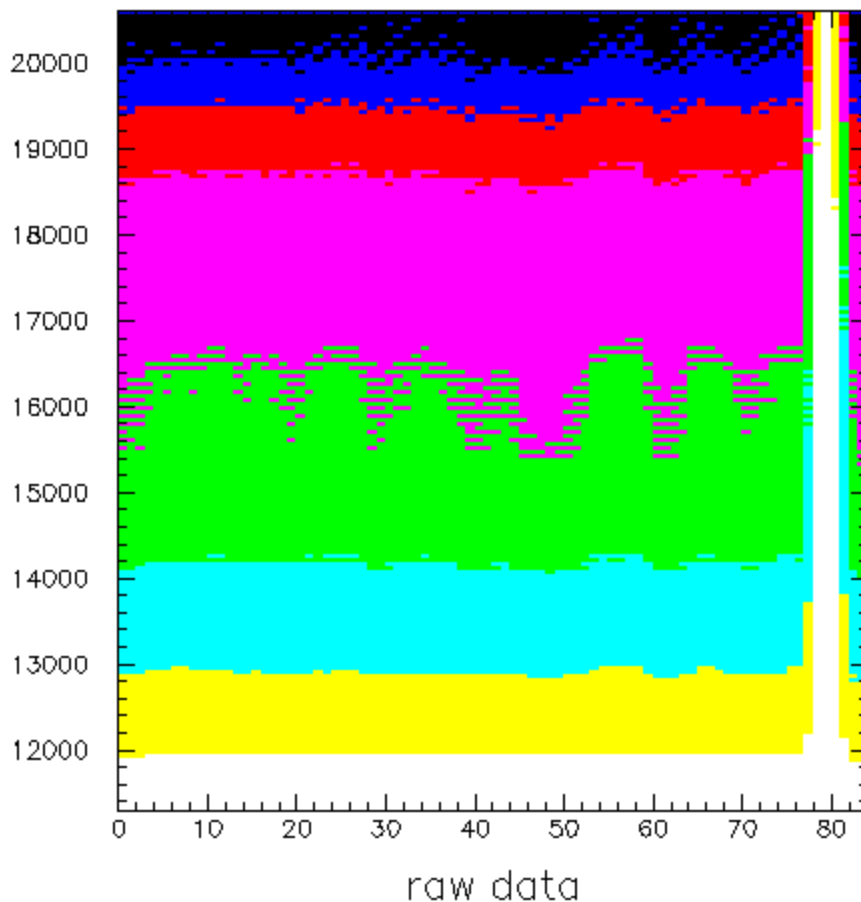
- read damper's fifo.dat w/ scipy
- calculate filter output
- make nice graphs
- streamline data crunching/graphing
- speed up readout

- get kicker DAC output going again

Here is what is currently on the to-do list on my board:

- collect data on 17 cycles
- reduce bloat
- check kick timing w/ scope readout
- check kick timing w/ beam feedback
- cook up notch finder to generate consistent bucket numbering
 - look for bucket w/ $\text{abs}(\text{Quadrature}[]) < 128$ counts on several turns
- allow Fast Time Plots of several intermediate stages' data, $\sum_{i=0}^{83} |x_i|$
- add threshold on Quadrature[] signal, so that empty buckets are clearly seen in filter output

Hey, OK! Notch finder works. I can consistently get the notch to appear in bucket number 79 on my FIFO readout graphs. With this working, I don't think I need to suppress the kick for low-pulse-height buckets, because I can simply use the existing feature to kick only a range of bucket numbers.



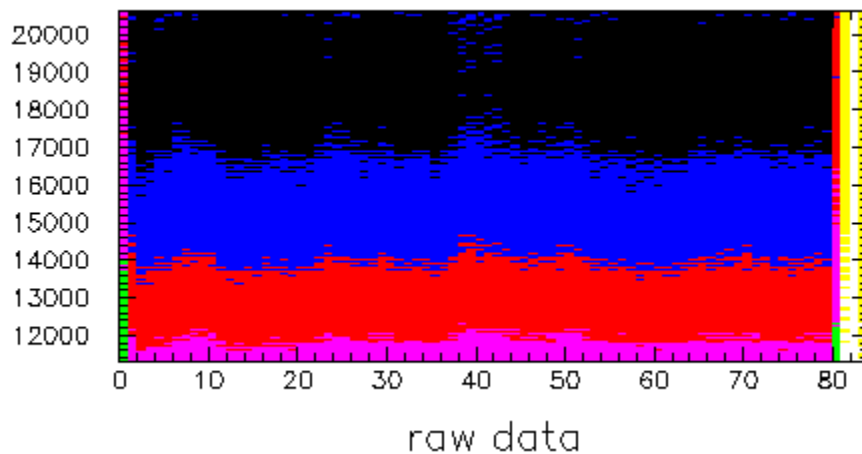
The priority for tomorrow is to look at the kick timing by reading out the oscilloscope. Then I'll use \$17 cycles and try to excite (antidamp) a small range of bunches.

2008-07-08

I added 3 to the bucket number in the fifo readout, so that the notch appears in the last few buckets. I am adding registers D4YKFB, D4YKLB, D4YGFB, D4YGLB, D4YIFT, D4YILT, which existed in the September 2006 firmware but were lost in my haste to move my computer up to the APC. KFB and KLB replace some of what was handled by the bunch-by-bunch RAM, providing first and last bunch numbers for gating the kick on and off. I am not sure what GFB and GLB did in 2006, but now I am using them as first and last bunches for which to output a saturated kick (independent of input signal), for adjusting timing. IFT and ILT are first and last turns on which to damp. Also, I removed the 17-clock pipeline delay for the Quadrature[] signal in iq_logic.tdf, which had been used for making 53 MHz and 2.5 MHz

timing line up in the MI damper.

Haven't broken anything yet. (This reminds me that I need eventually to have a solid set of diagnostics that do not require beam to test, e.g. a second board providing simulated signals.)



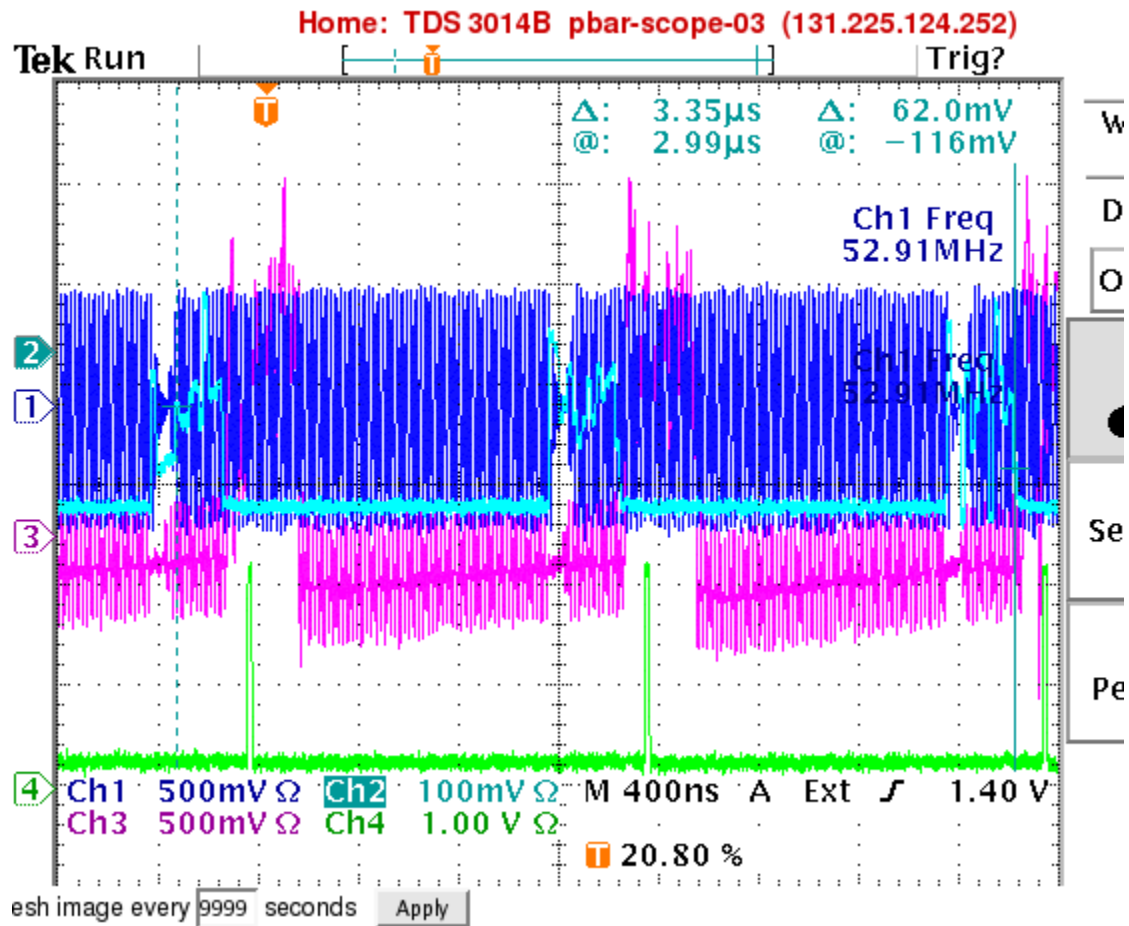
Crude check of timing using oscilloscope doesn't look too bad. I think I will be at most a few buckets off. There is a delay table (in units of 1/8 RF cycle) vs. high bits of turn number, in `damper_kickout_logic.tdf`. I have not yet filled this table. I think I will need to write offsets corresponding to $O(100\text{ns})$ of delay. 100ns corresponds to a difference of 1.5 buckets over the 38 MHz to 53 MHz slew of the Booster.

Bill Pellico tells me that he has \$17 cycles set up to look like normal cycles, including the notcher. He tried it last night up to 10 turns. So I have the go-ahead!

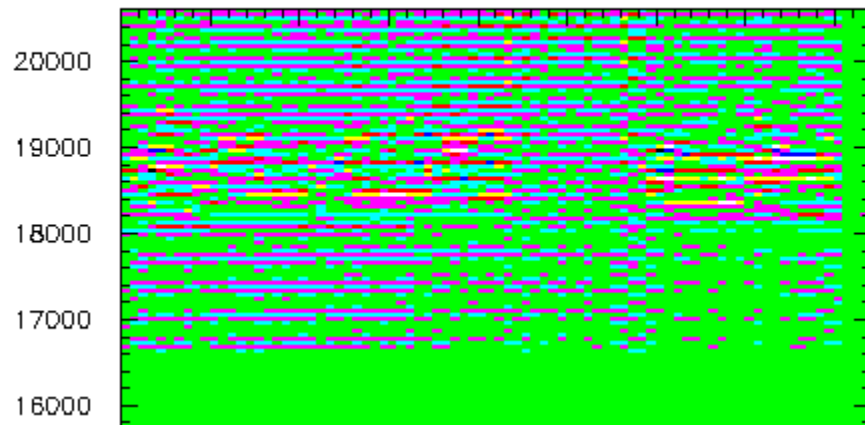
Am working with \$17 cycles. I am first trying to figure out how to get the feedback to kick the same bunch that it is measuring.



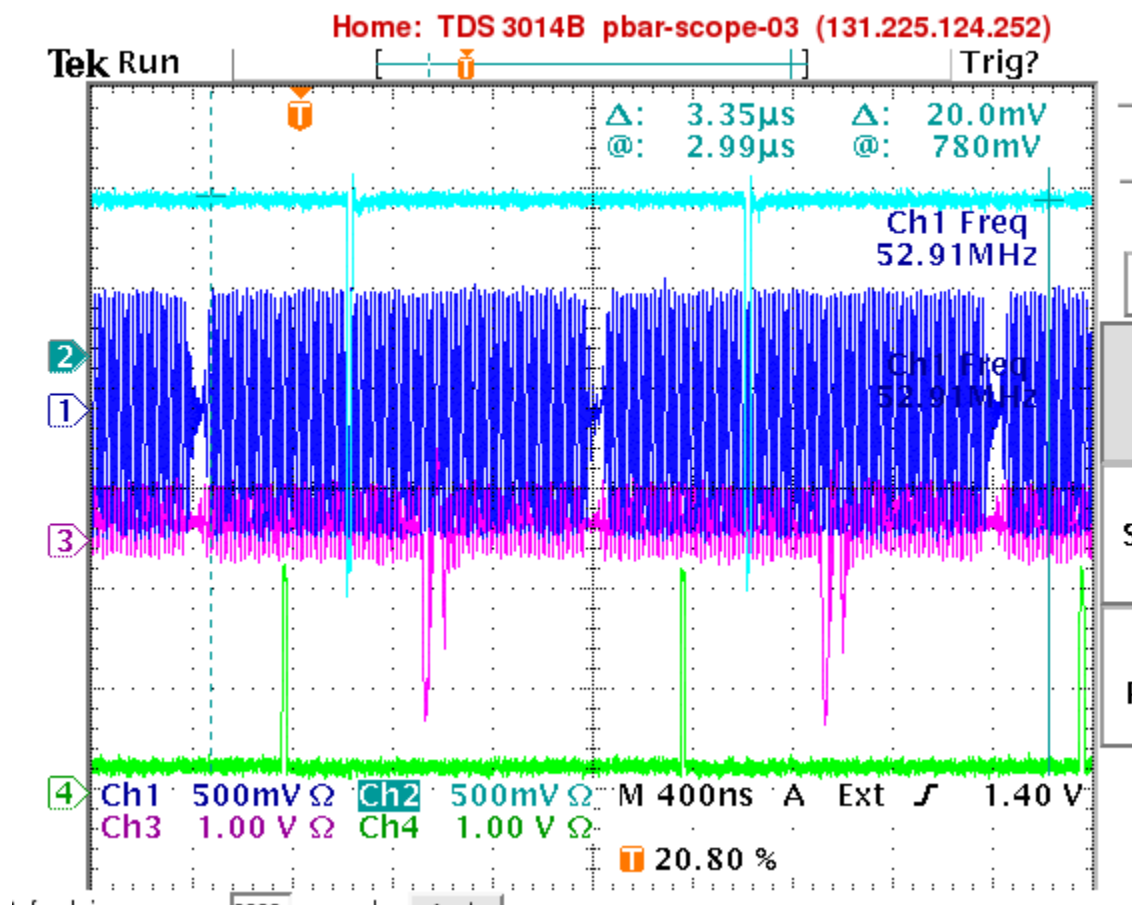
graph01.png 16:01



graph02.png 16:21



graph03.png 16:29



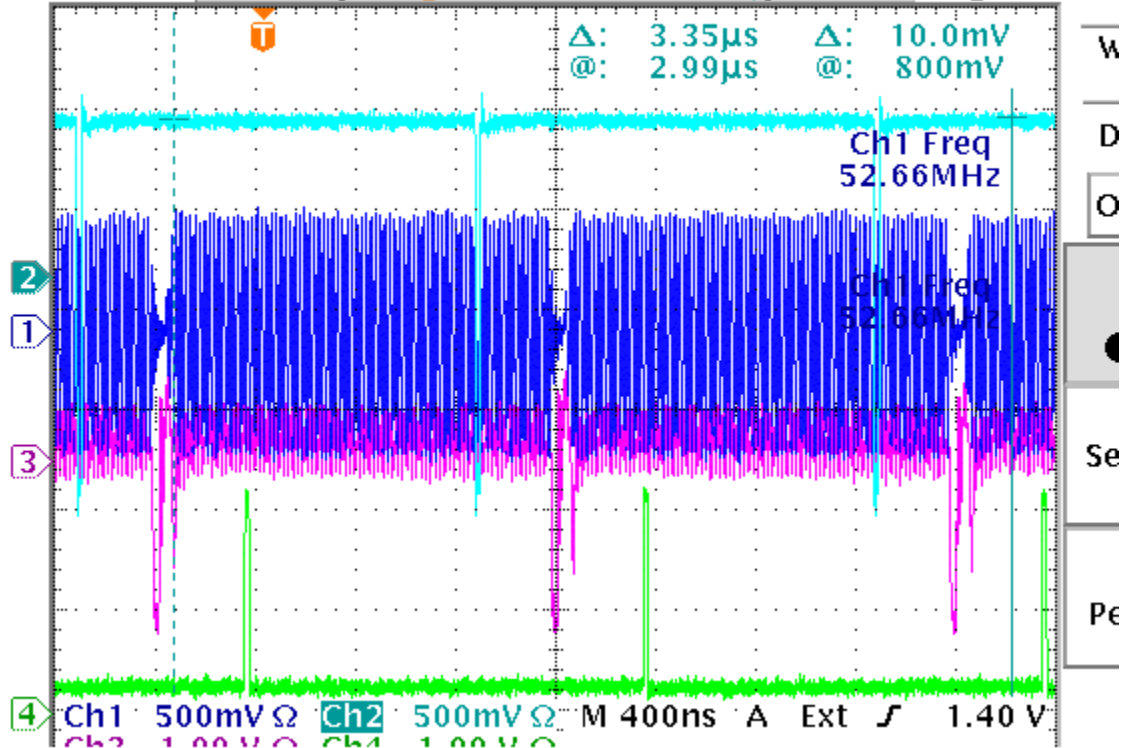
graph04.png 17:08

After a bunch of false starts, I think I found a pretty good method. I set D4YPHA:=0 and D4YFIR:=0x707, which means that the kick signal is proportional to the beam signal (no subtraction). Then I can look for the notch in the kick signal (and the fanback signal). Things should be properly aligned when the fanback signal shows the two notches overlapping. Right now (17:10) graph04.png shows the notch in the kick coming about 35 buckets before the notch in the beam. OK, after inserting 35-bucket delay, graph05.png looks much more closely aligned. Now let's see what feedback does. That's graph06.png (scope), graph07.png (3-turn filter). After flipping sign of feedback (graph08.png), I see a much larger effect.

Home: TDS 3014B pbar-scope-03 (131.225.124.252)

Tek Run

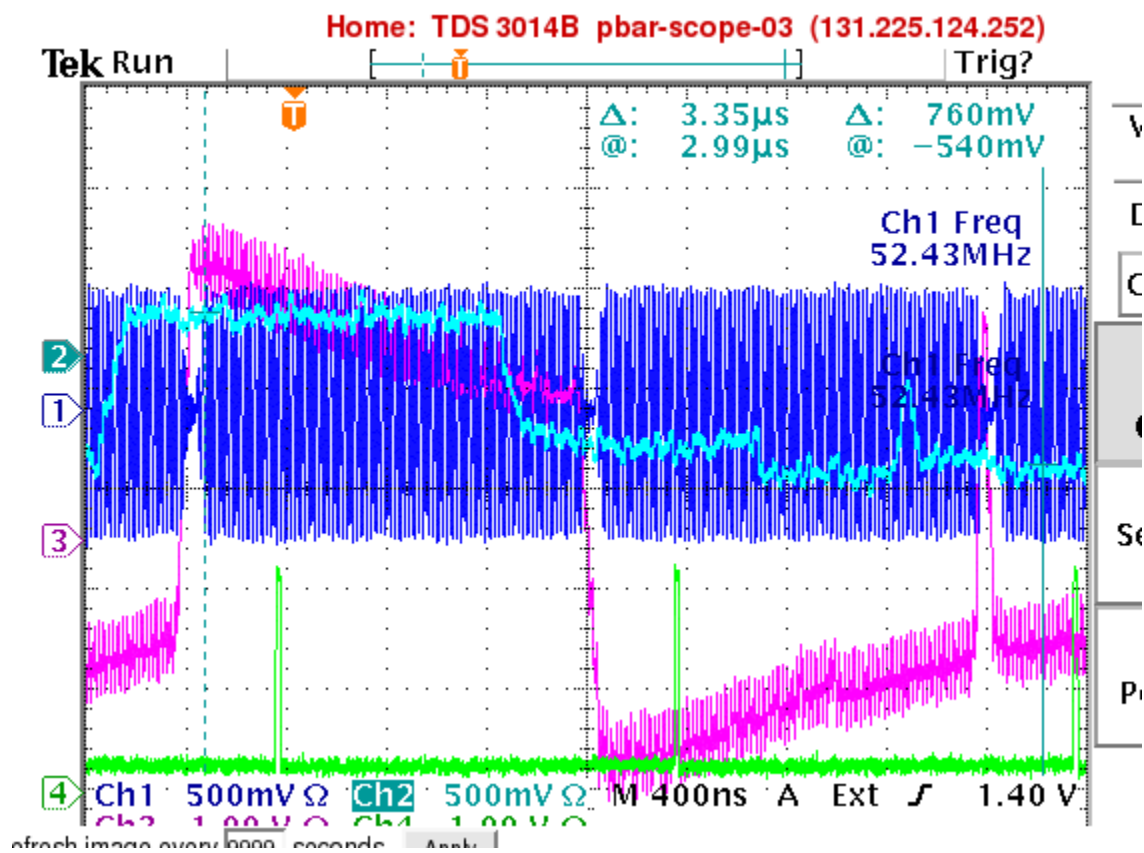
Trig?



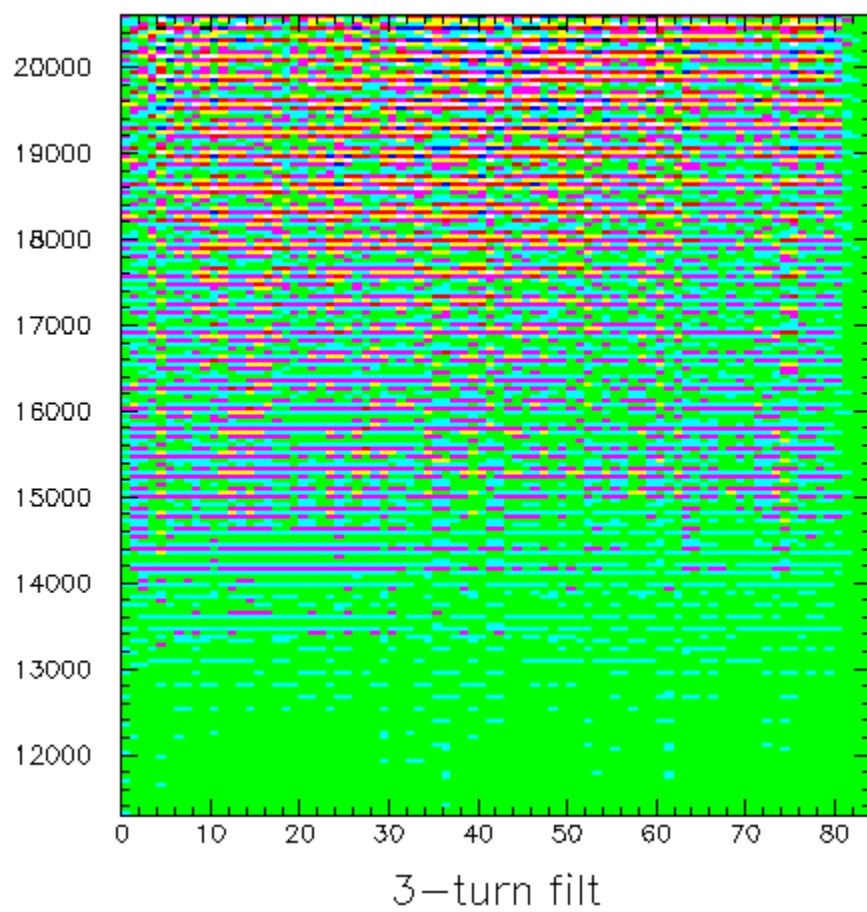
refresh image every 9999 seconds

Apply

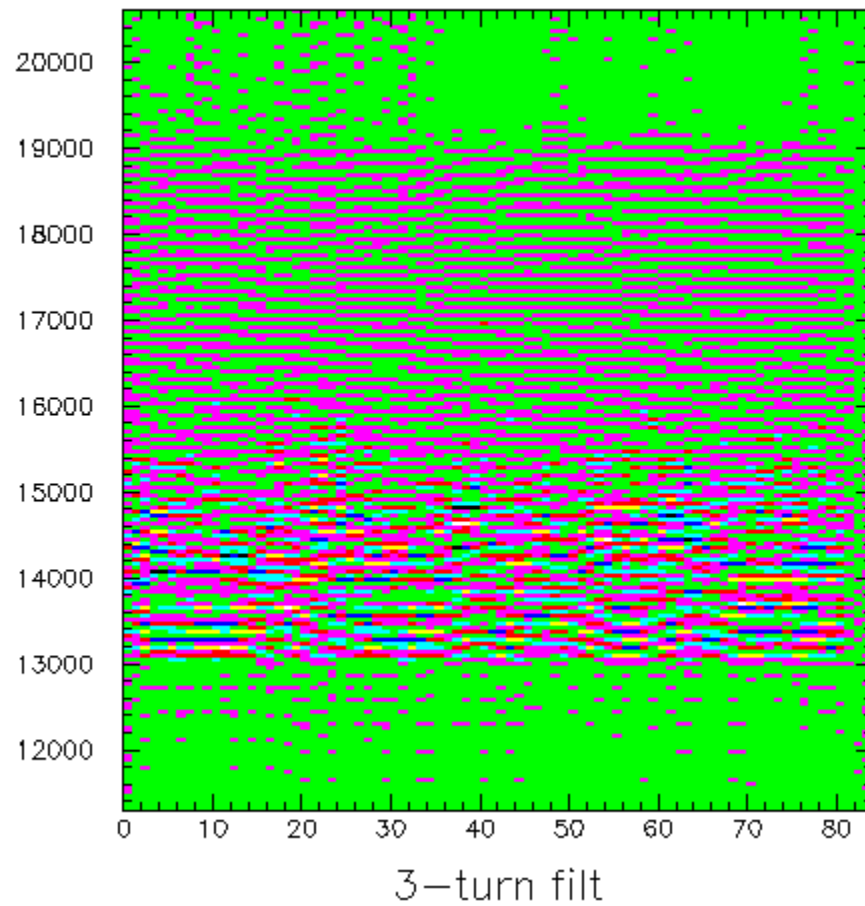
graph05.png 17:23



graph06.png 17:27

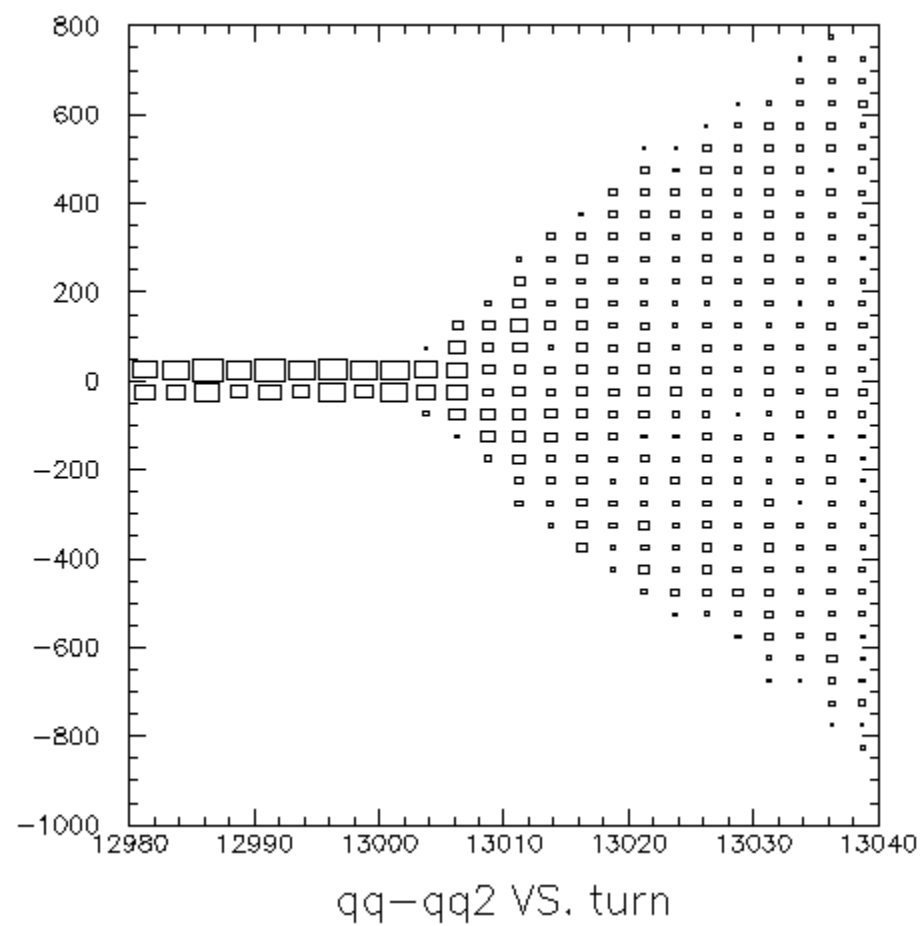


graph07.png 17:29

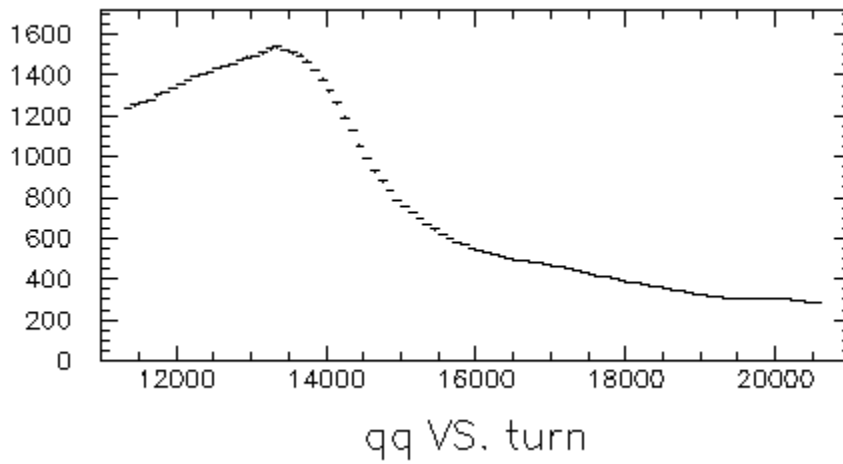


graph08.png 17:33

Impressive! graph09.png shows the 3-turn filter rapidly increasing once antidamping is turned on at turn 13000. graph010.png shows beam loss starting around turn 13500.

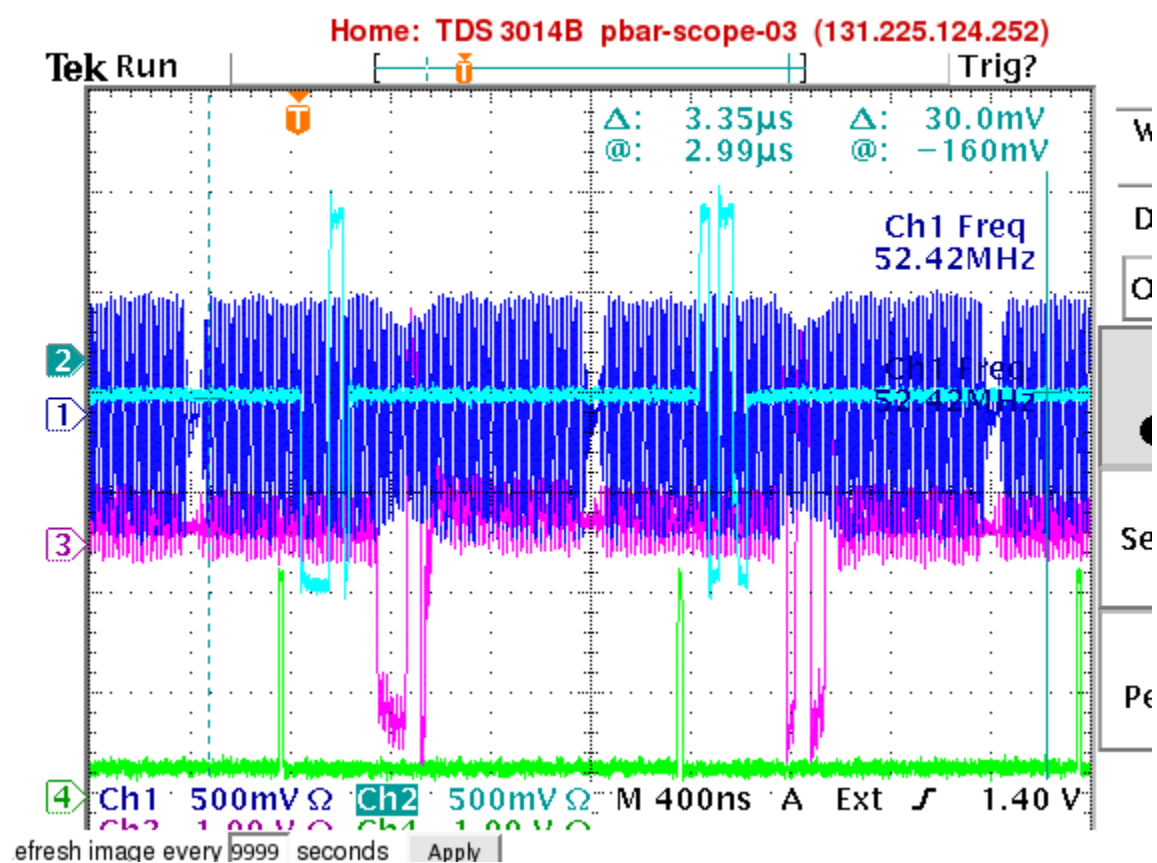


graph09.png 17:38

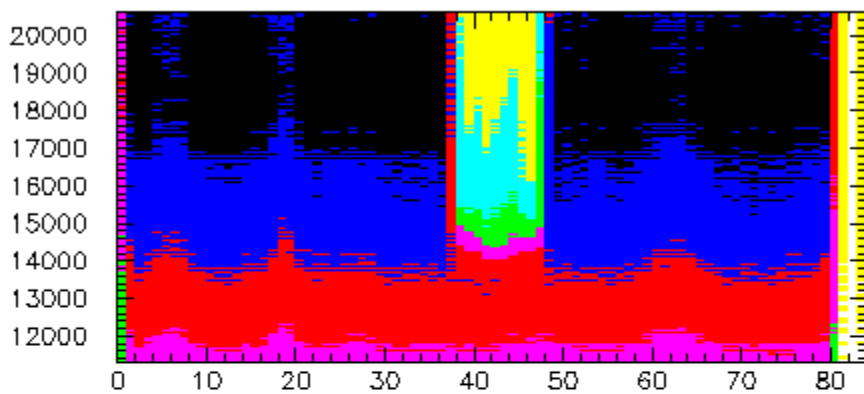


graph010.png 17:40

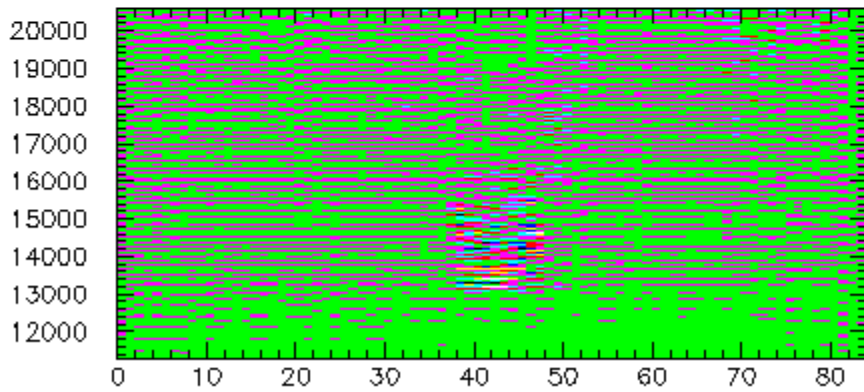
Now just enable (anti)damping for buckets 40-50. graph011 is scope readout at turn 14000. graph012.png shows that we managed to antidamp bunches 38-48 out of the machine! graph013.png zooms in on bunches 38-48 in FIFO data. graph014 shows bunches 43 and 44 blowing up.



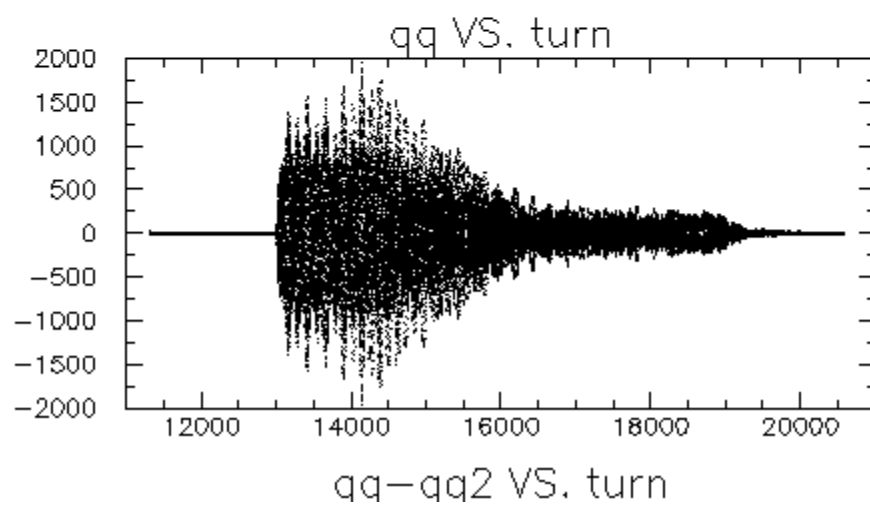
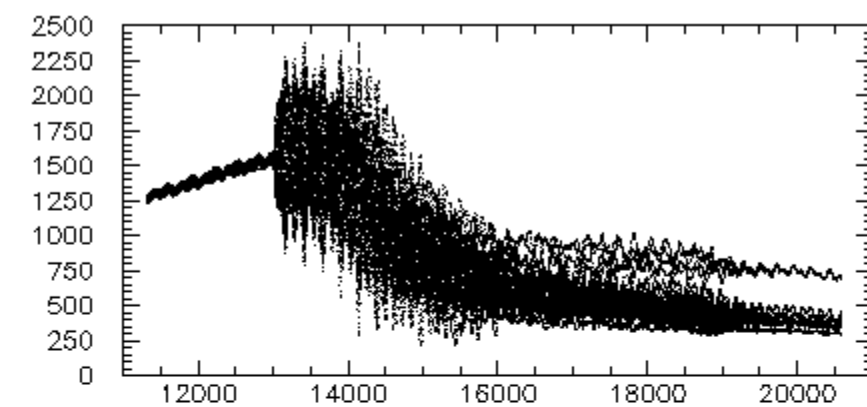
graph011.png 17:43



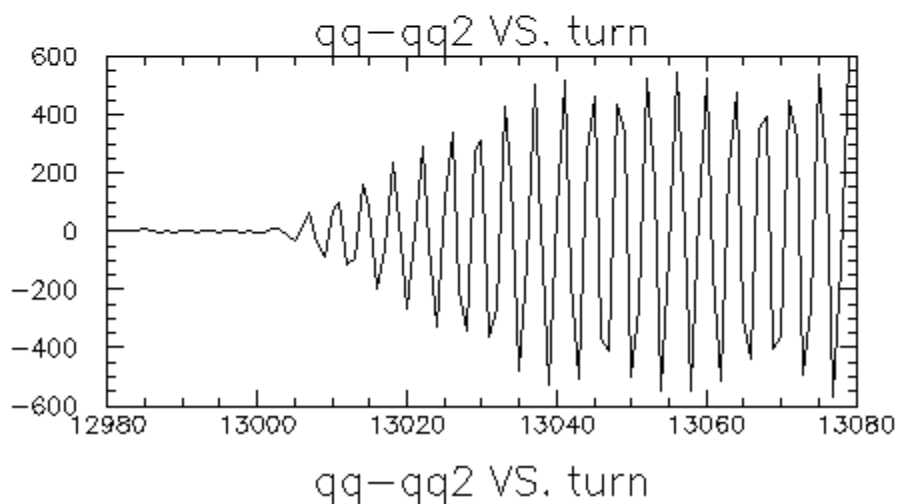
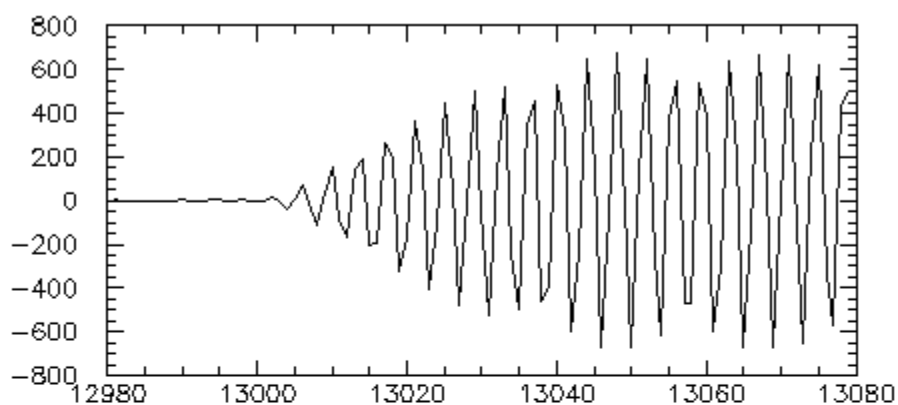
raw data



3-turn filt

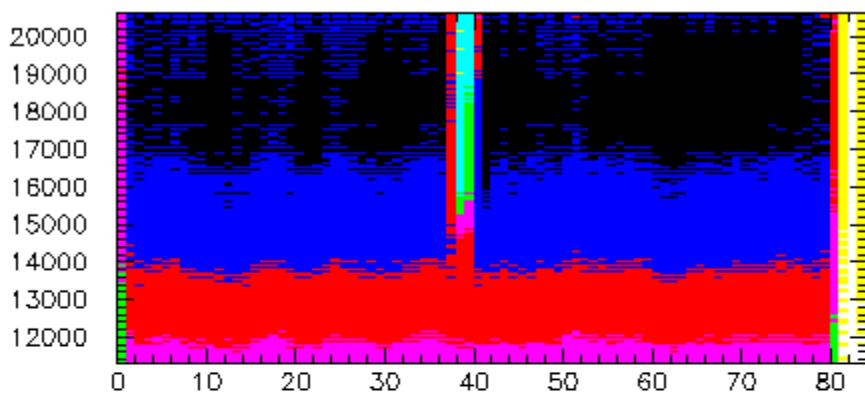


graph013.png 17:49

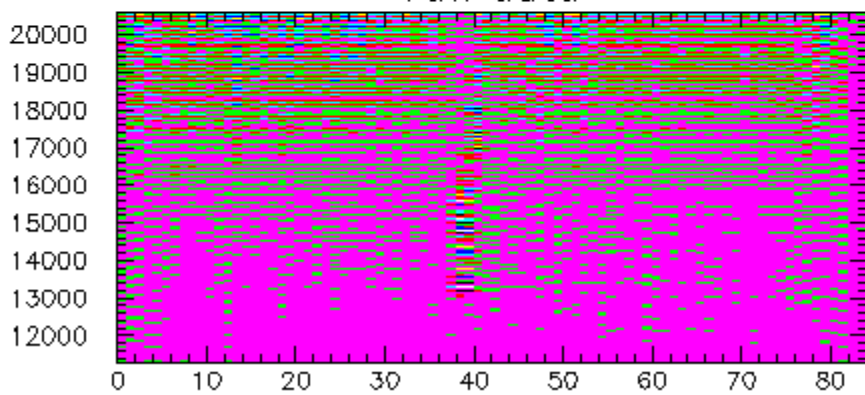


graph014.png 17:53

graph015 shows feedback on just two bunches (40-41). now just doing bunch 20, so that fanback shows kick signal near scope trigger: graph016. graph017 shows timing (zoomed in a bit) for feedback on only bunch 23. scope triggers at turn 13100. By the way, for this antidamping, I:D4YPHA=22. Amplifier gain has been at 100% all shift. graph018 shows that the kick signal should be about 1/4 turn earlier. Asking the board to (anti)damp bunch 23 causes the (anti)damped bunch to show up in bucket 21 in my fifo readout, according to numbering inserted by dofifo.py (as of 18:40). graph019 shows antidamping from turn 13000 to 13025 and then damping. It looks as if the damper S/N needs improvement, as the induced oscillation is not damped all the way back down.

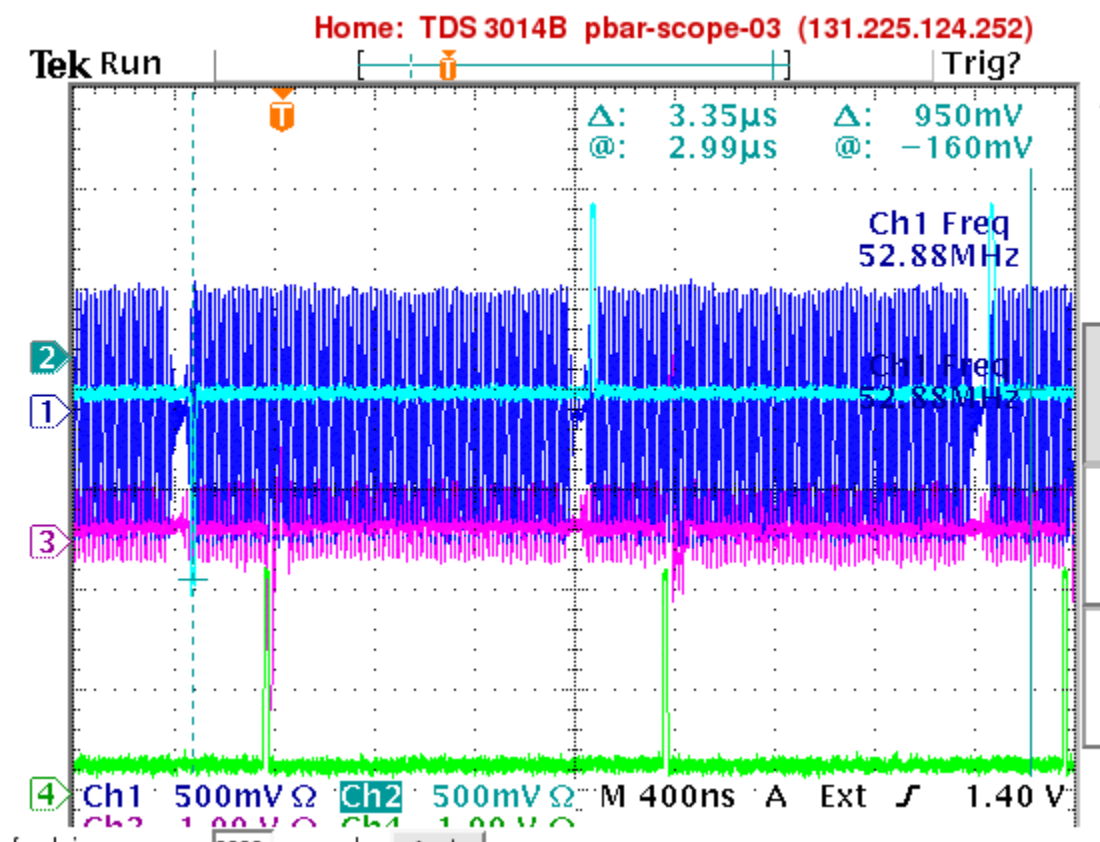


raw data

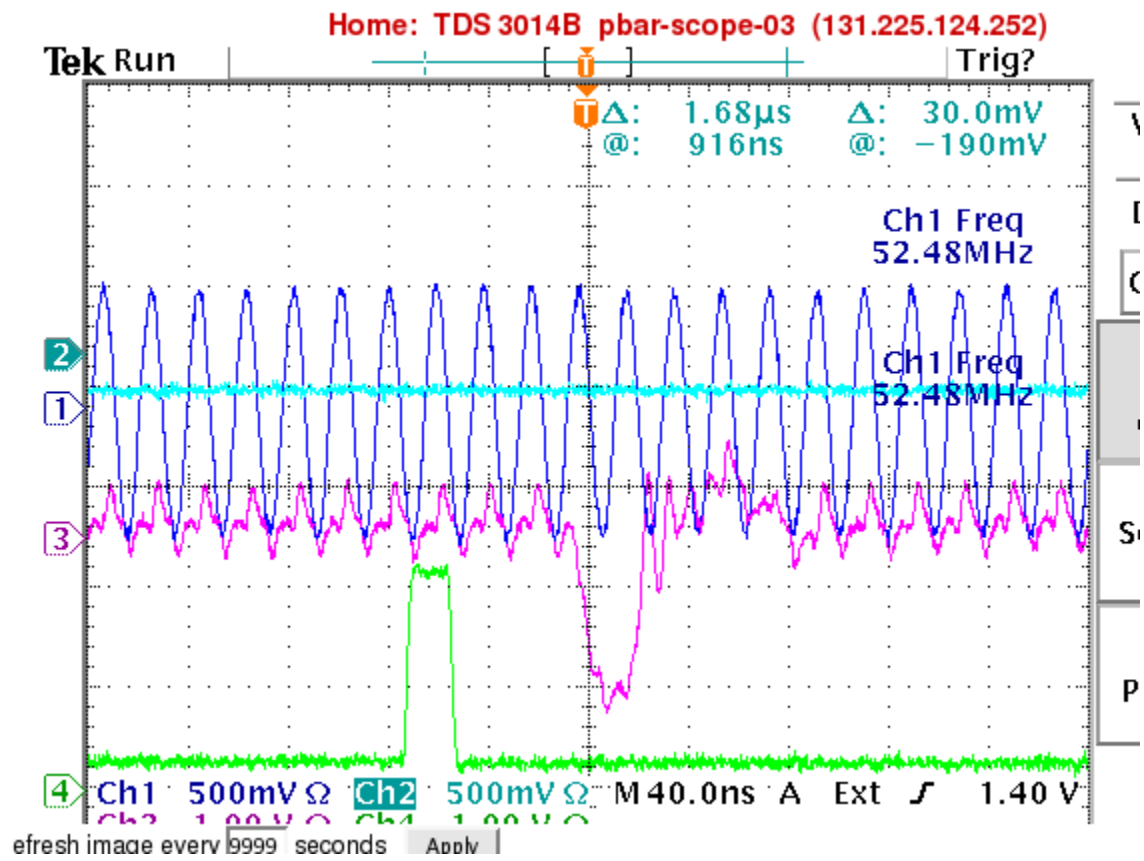


3-turn filt

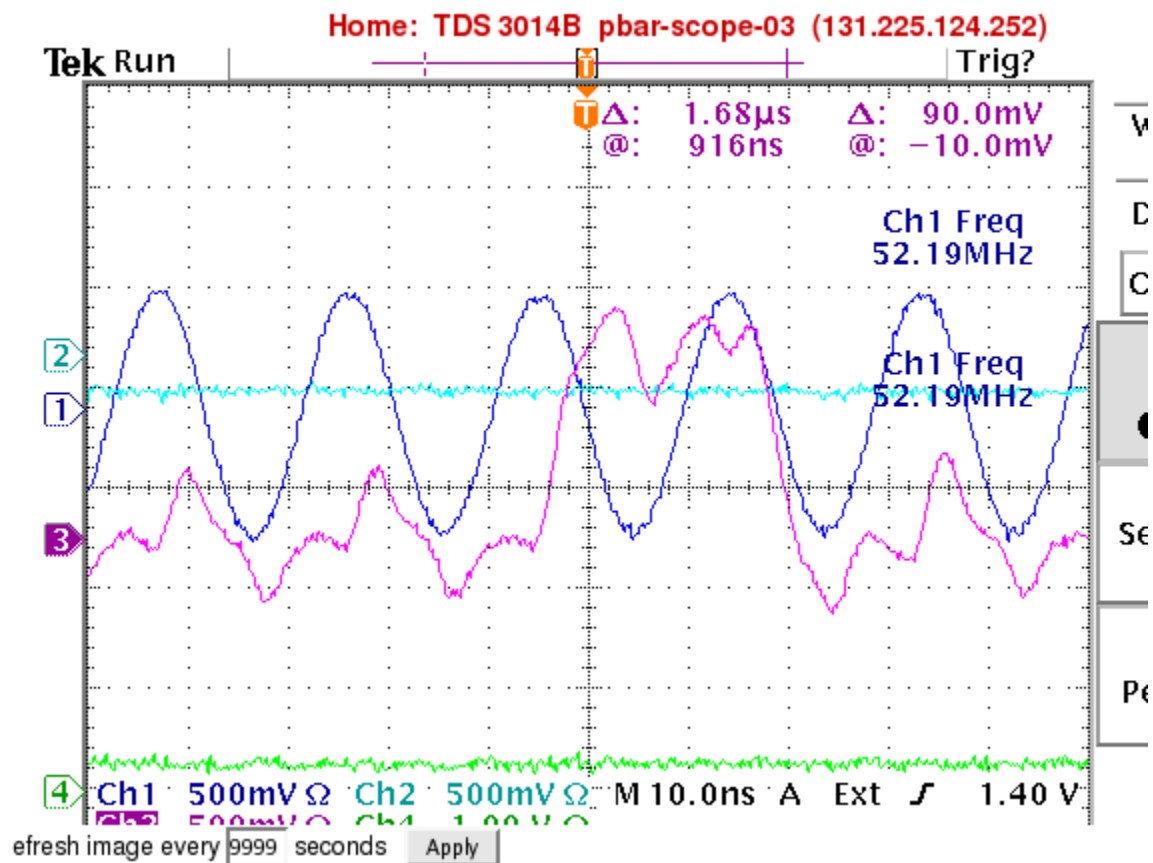
graph015.png 18:01



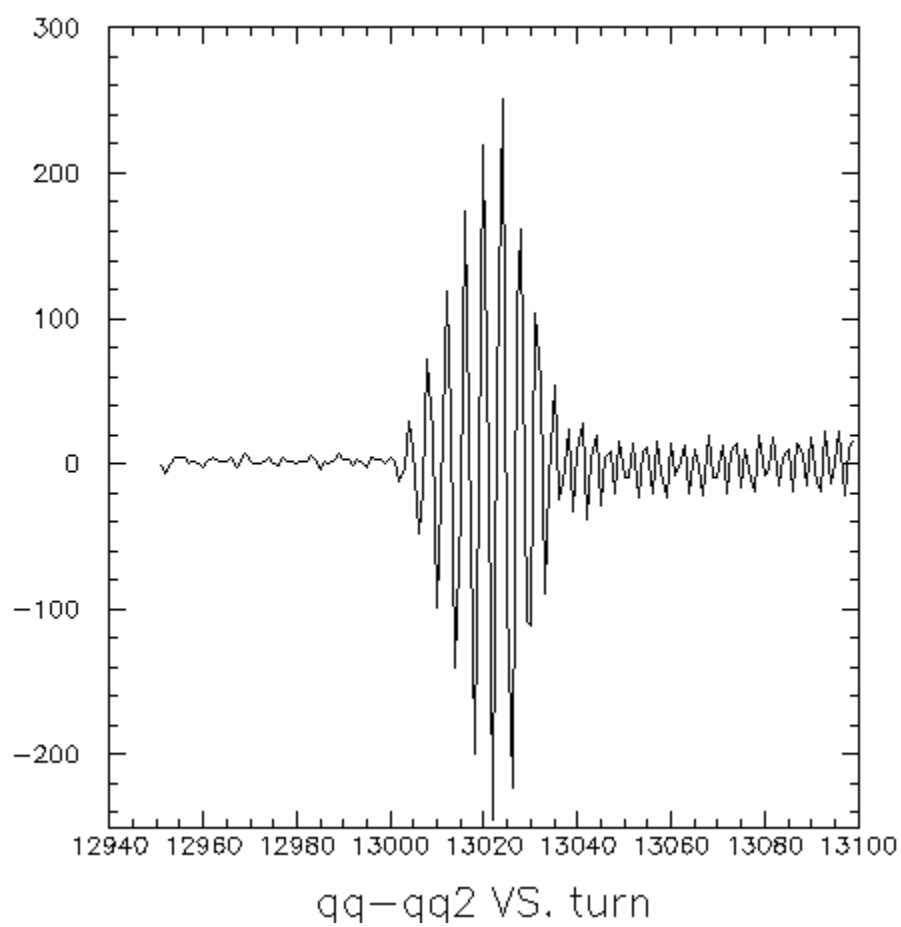
graph016.png 18:12



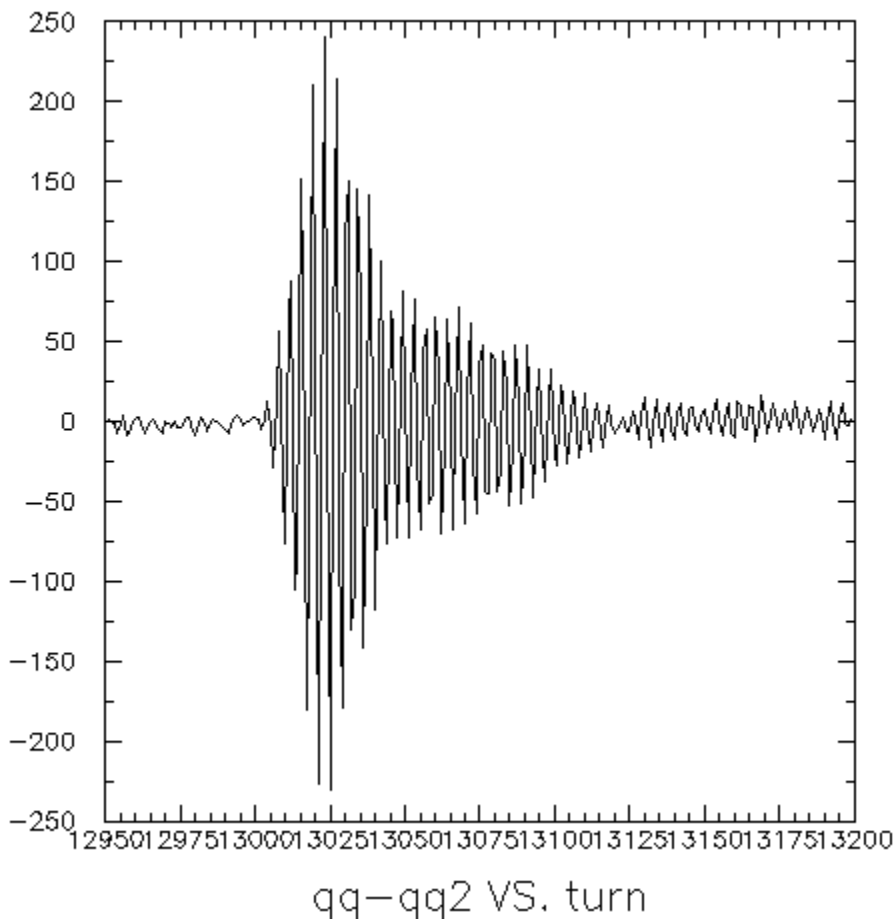
graph017.png 18:24



graph018.png 18:38

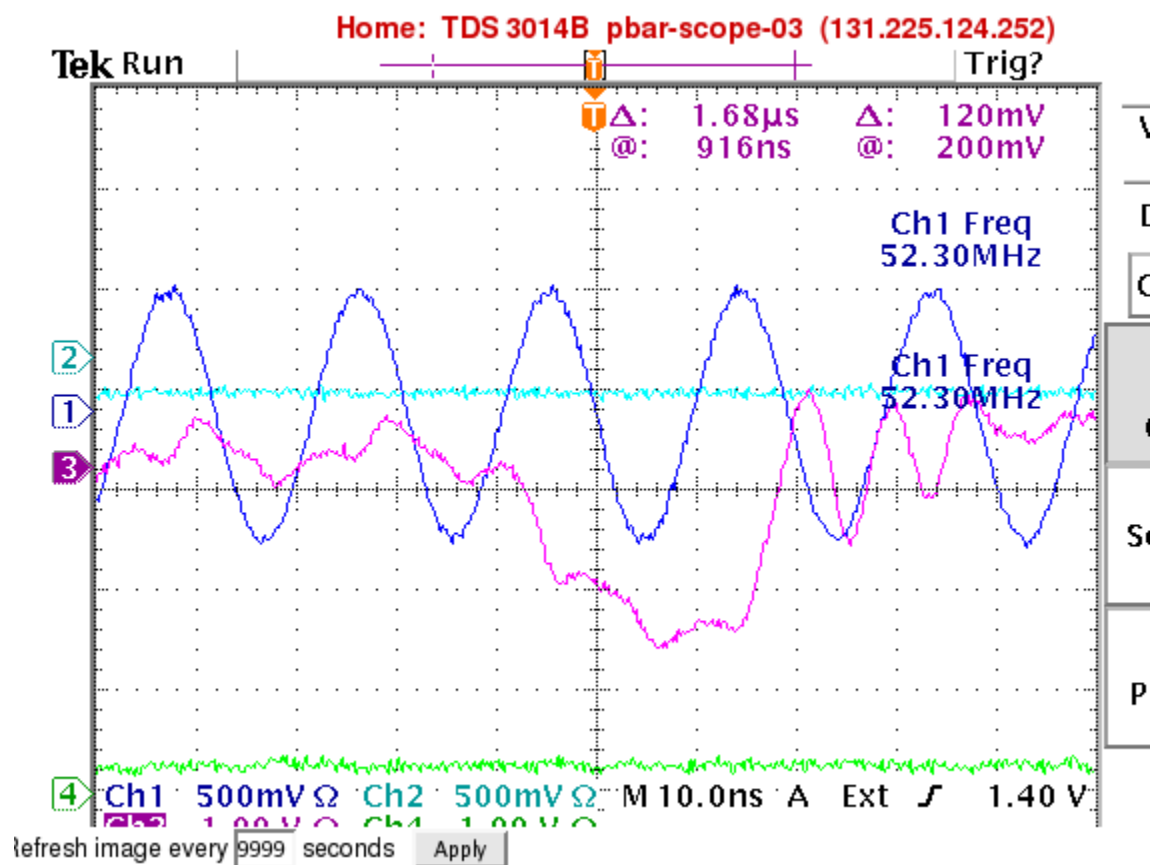


graph019.png 18:43



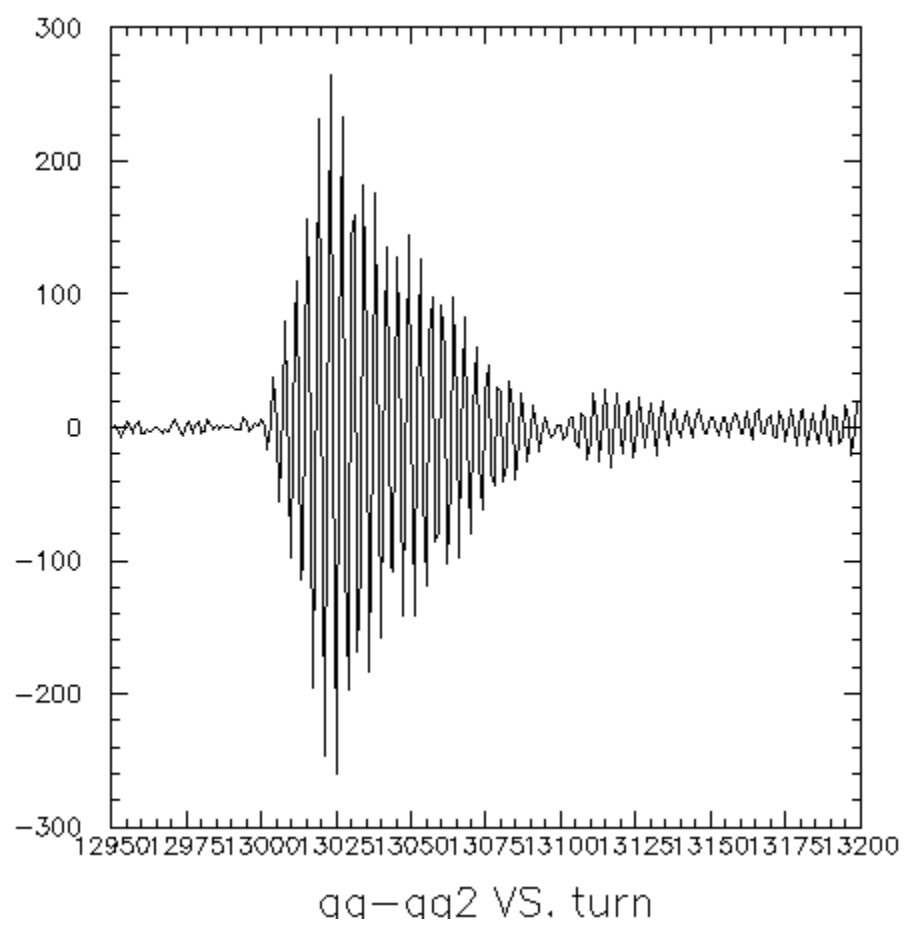
graph0110.png 19:02

graph0111 shows timing adjusted back 1/8 bucket. I think this will be easier with the higher-bandwidth oscilloscope (which has no web display, but I can read out the traces over the network). The oscilloscope we see here has just 100 MHz analog bandwidth. Should put this on to-do list: look in detail at beam and fanback signals with fast scope while damping.

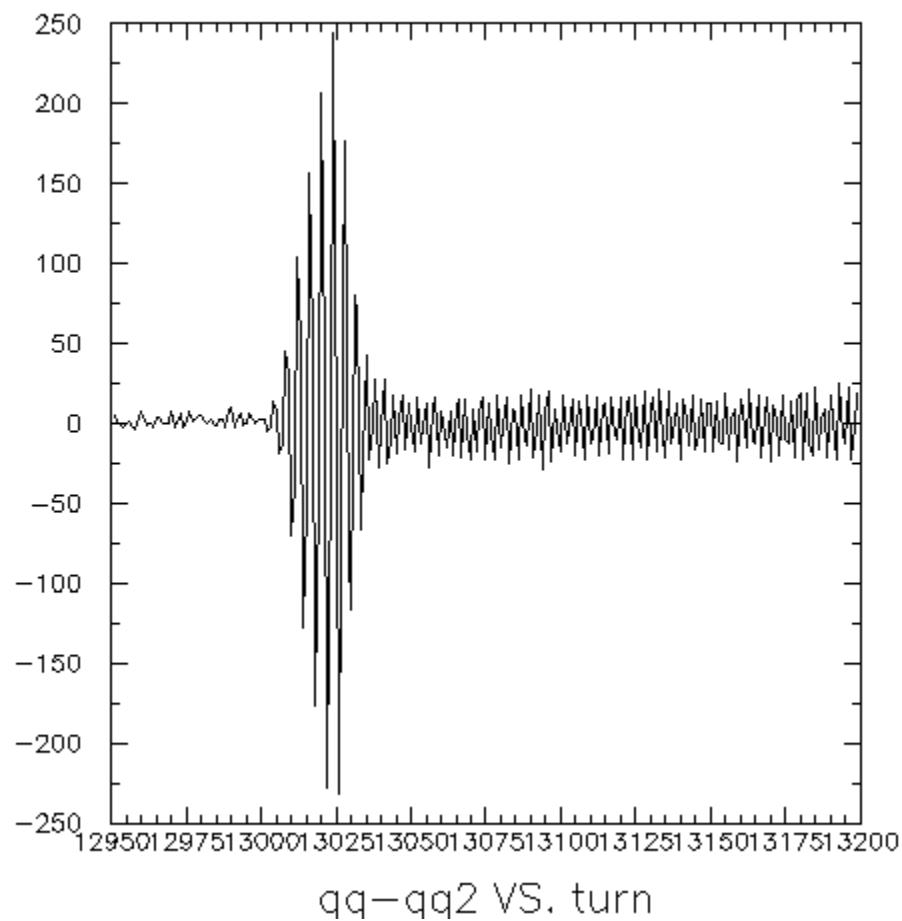


graph011.png 19:03

graph0112 shows antidamping 25 turns followed by turning off damper. graph0113 shows antidamping 25 turns followed by damping.

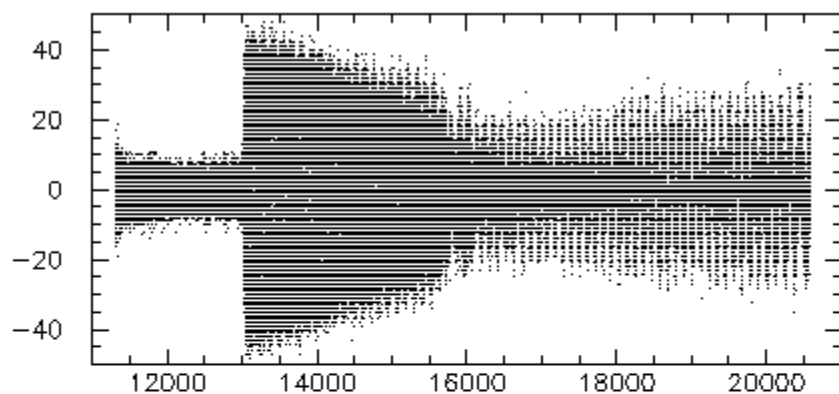


graph0112.png 19:06

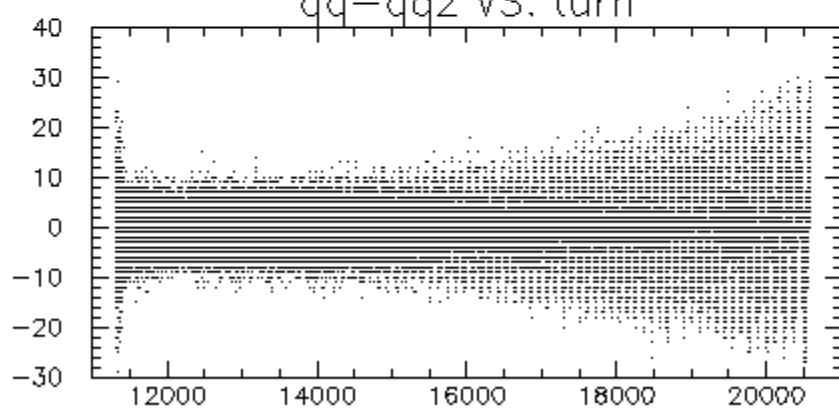


graph0113.png 19:10

now just damping. graph0114 shows 20 bunches being damped, 20 bunches not being damped, for gain=15. next, try gain=3. (maximum gain is 31.) graph0115 has gain=3. graph0116 has gain=5. graph0117 has gain=7. Clearly the damper is injecting noise, and does not seem to damp the naturally occurring oscillation toward the end of the cycle. Hmmm. This is becoming a bit mindless. It's time to step back and think a bit. I do want to check the timing tomorrow by digitizing the fanback signal with the fast oscilloscope.

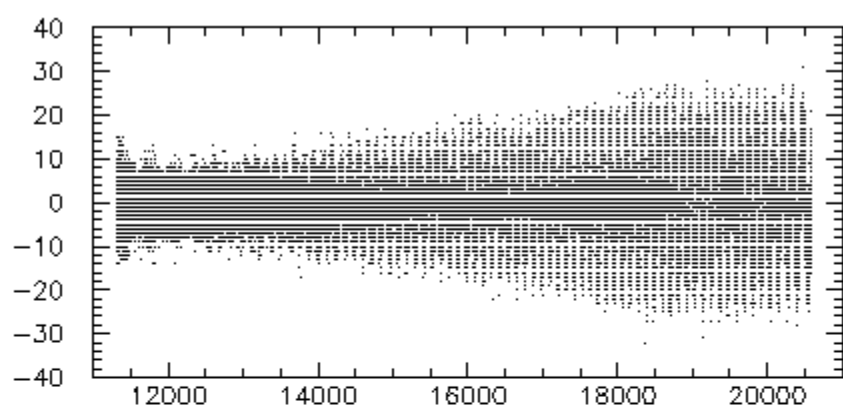


$qq - qq2$ VS. $turn$

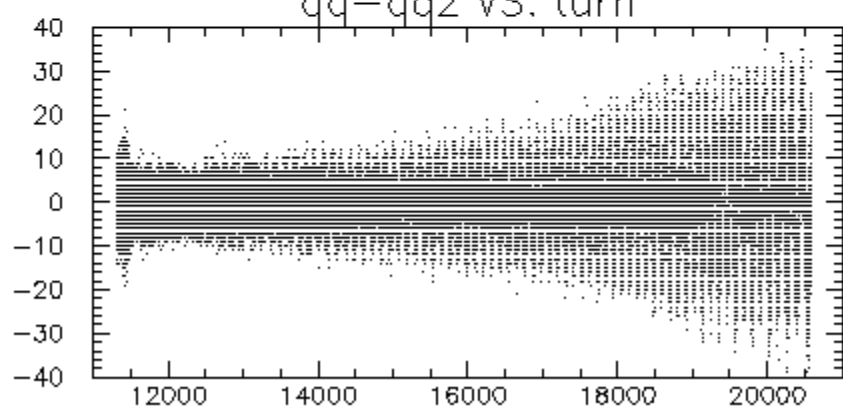


$qq - qq2$ VS. $turn$

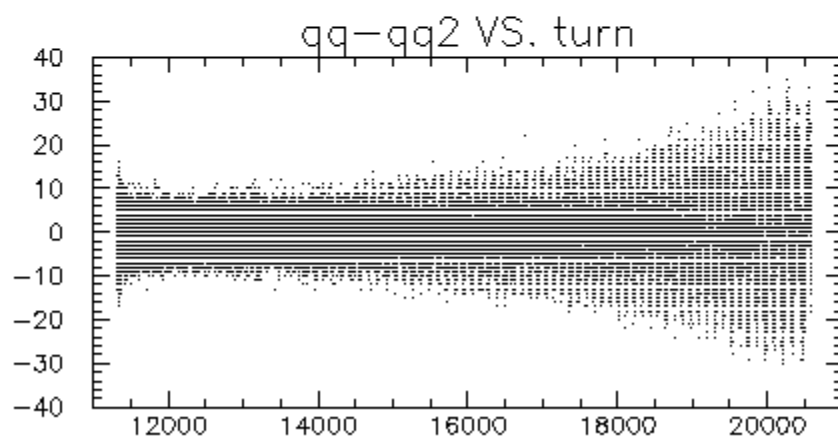
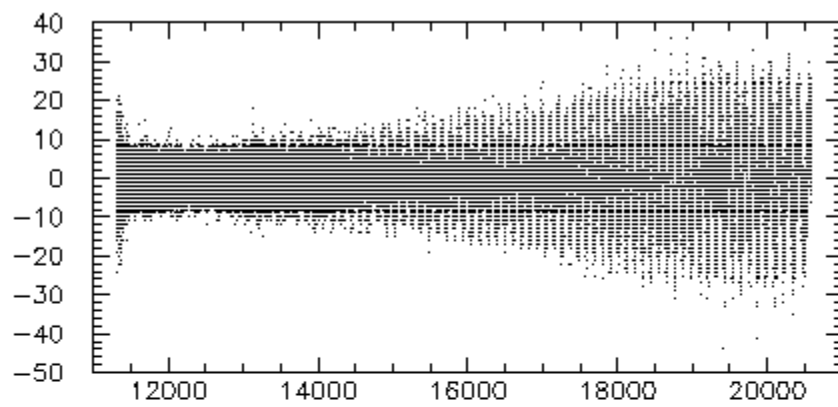
graph0114.png 19:30



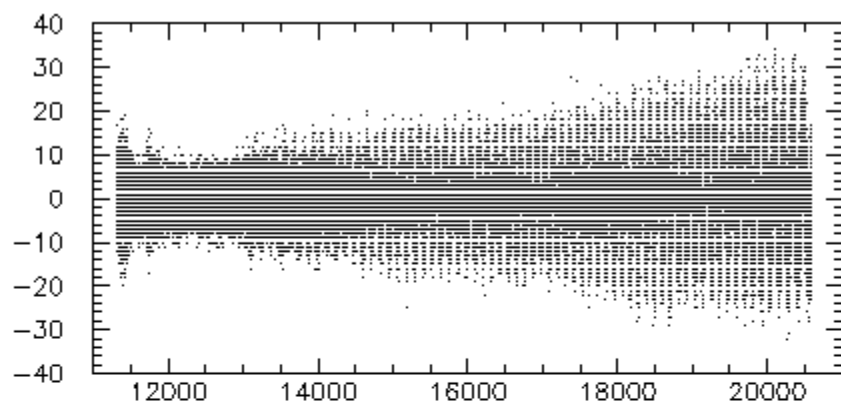
$qq - qq2$ VS. $turn$



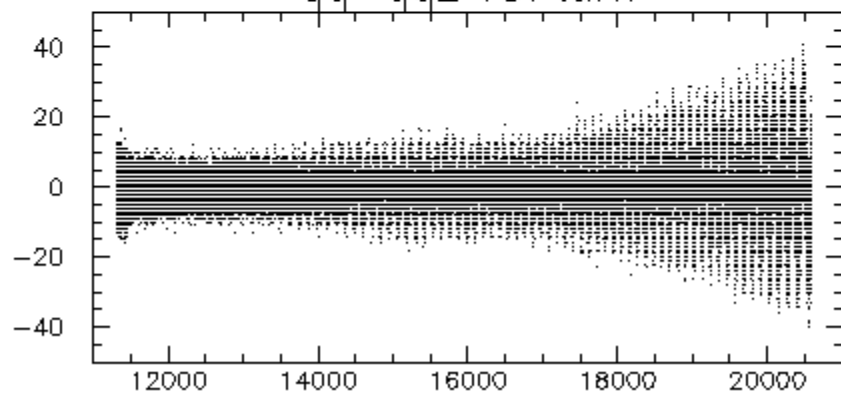
$qq - qq2$ VS. $turn$



qq-qq2 VS. turn



qq-qq2 VS. turn

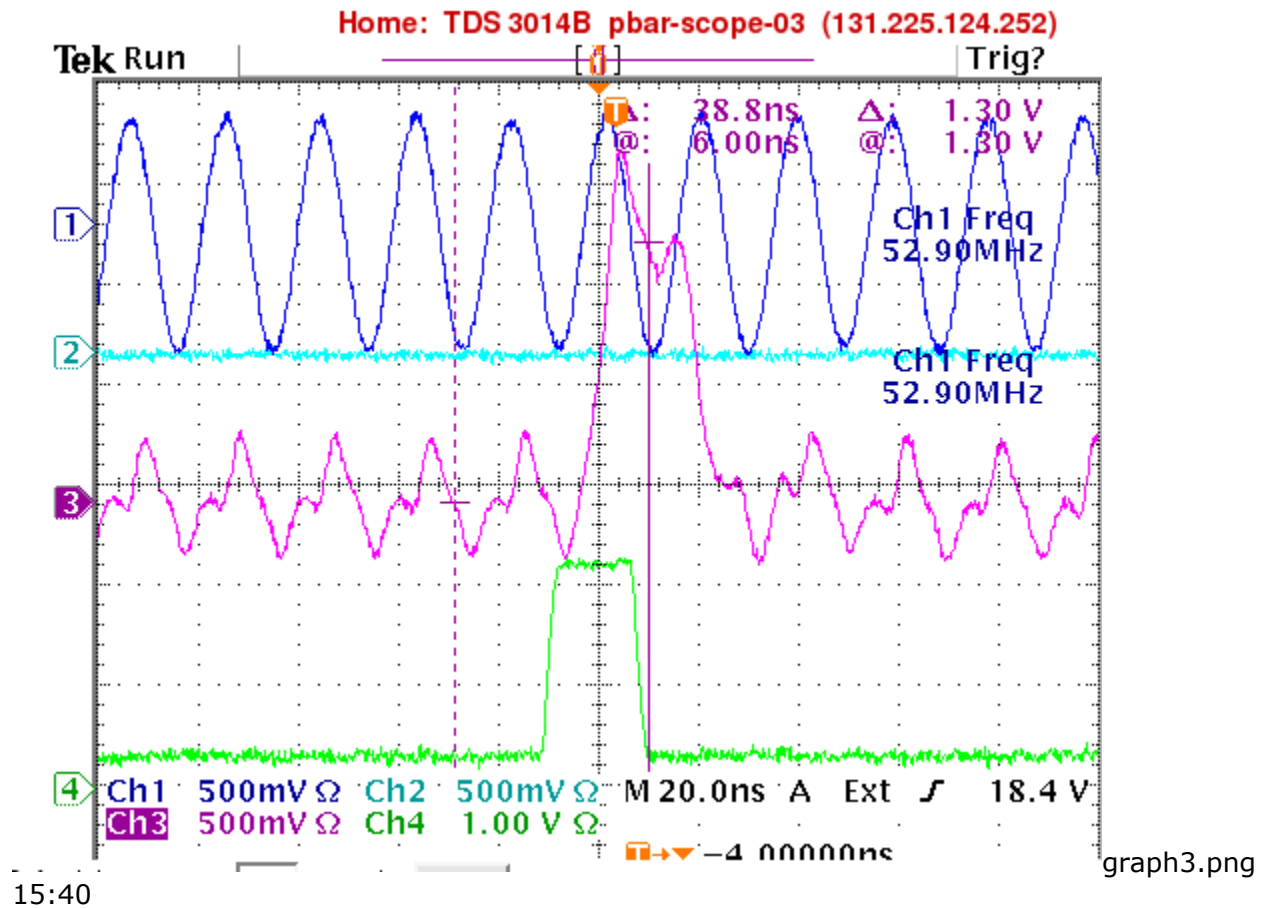


aa-aa2 VS. turn

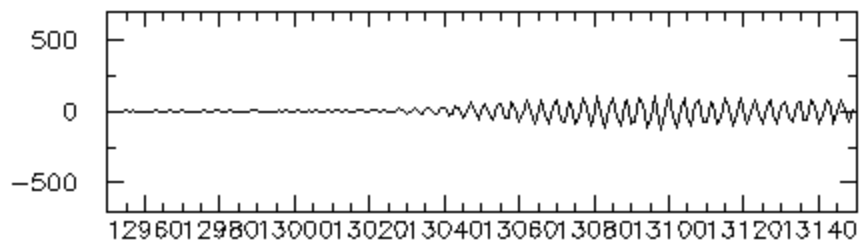
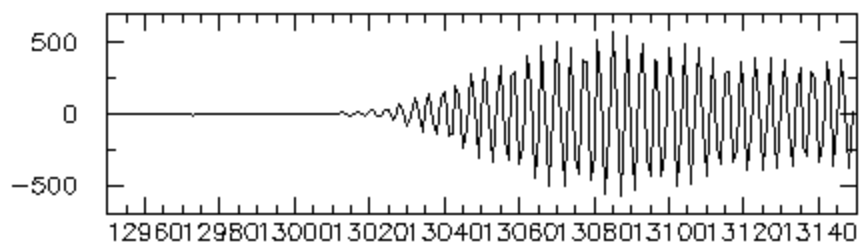
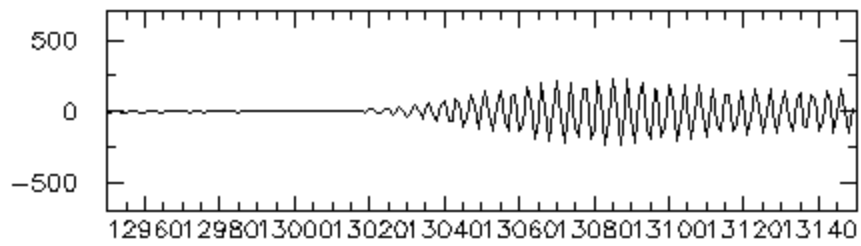
graph0117.png 19:46

2008-07-09

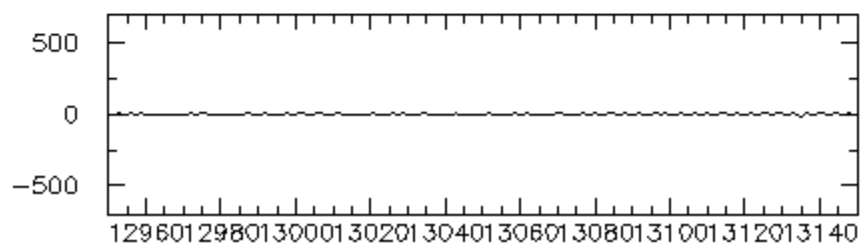
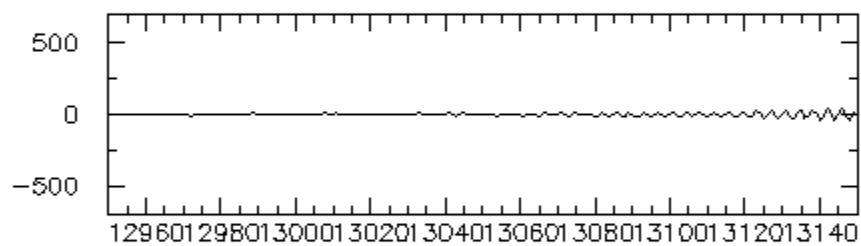
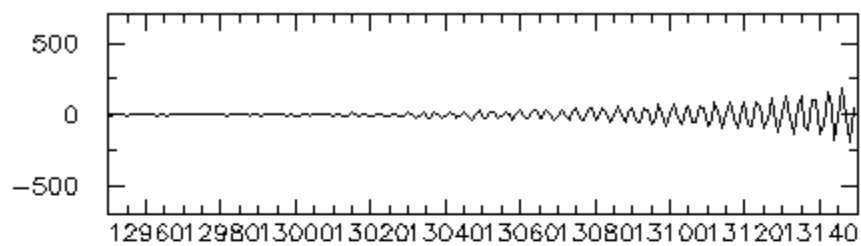
graph3.png (15:40) shows my attempt to center the kick on one bunch. the right-hand cursor is where I estimate a bunch should be centered. (the left-hand cursor is centered on a bunch.) as yesterday, this is the kicker fanback signal. I reduced the length of the kick length from a whole bucket to 6/8 of a bucket. (17:05 Aha! I didn't! I changed the program and started it compiling but never loaded it into the board! Kick is still a whole bucket long.)



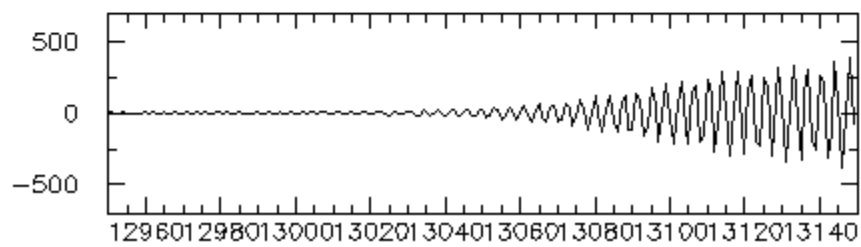
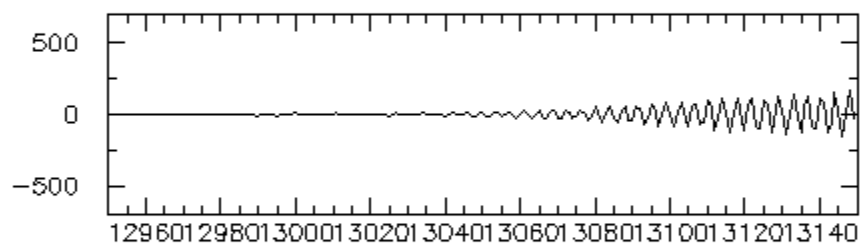
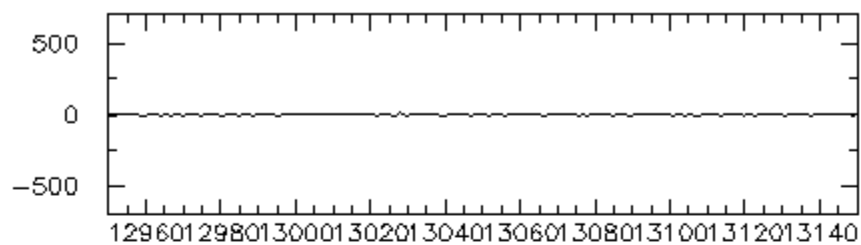
graph4.png (15:42) is bucket 17,18,19 from fifo (readout numbering scheme), when damper is touching only bucket 20 (damping numbering scheme). delay (in 1/8 buckets) is $35 \times 8 - 1$. graph5.png is bucket 17,18,19, etc., with $34 \times 8 - 1$. graph6.png is same for $36 \times 8 - 1$. graph7.png back to normal, and betatron phase moved 180 degrees. graph8 (16:05) is antidamping from turn 13000 to 13070 and then damping, still touching only bunch 20.



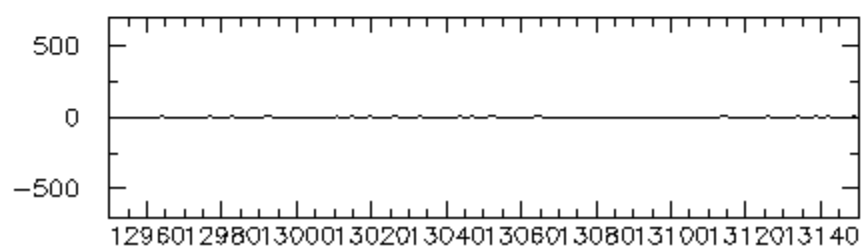
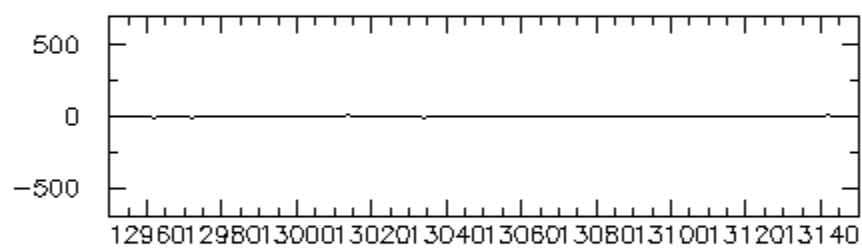
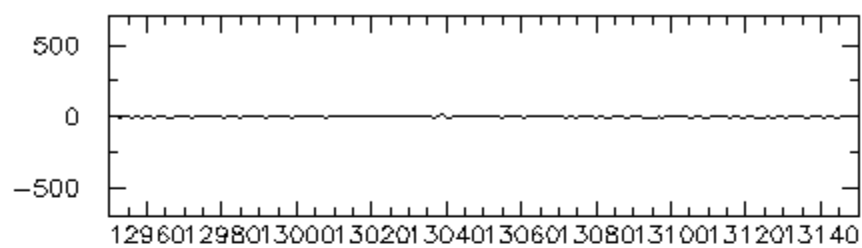
graph4.png 15:42



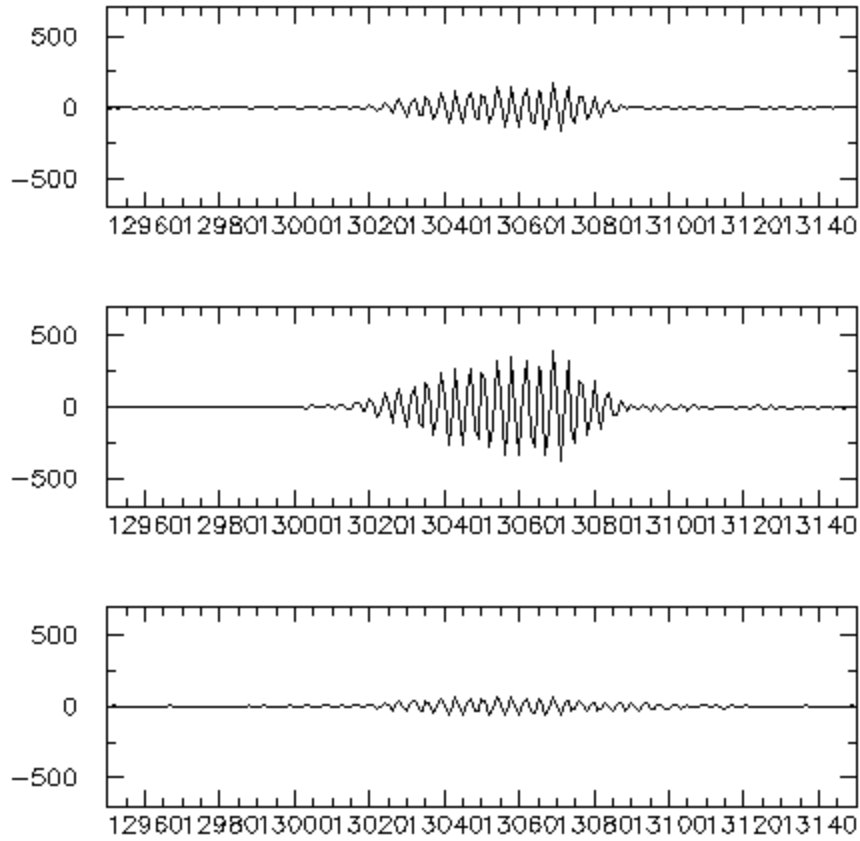
graph5.png 15:47



graph6.png 15:49

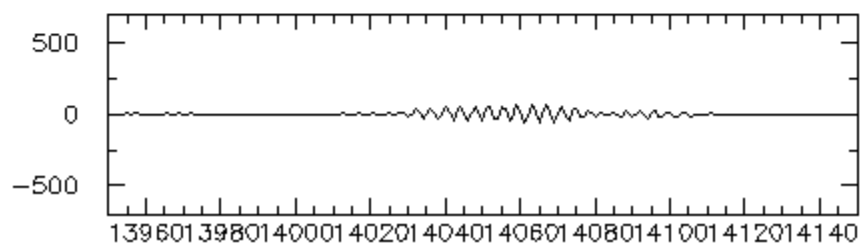
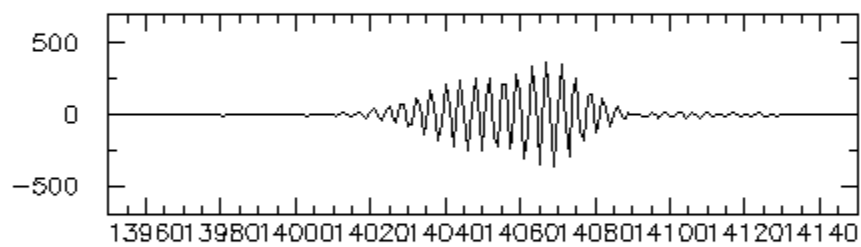
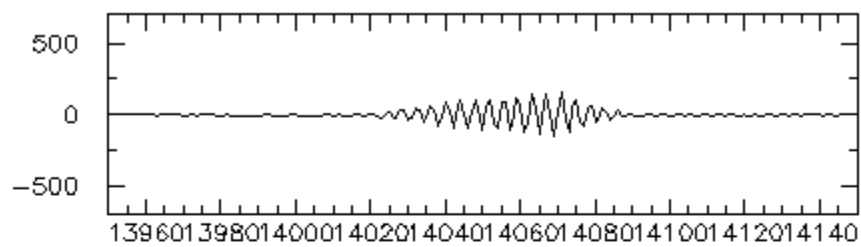


graph7.png 16:04

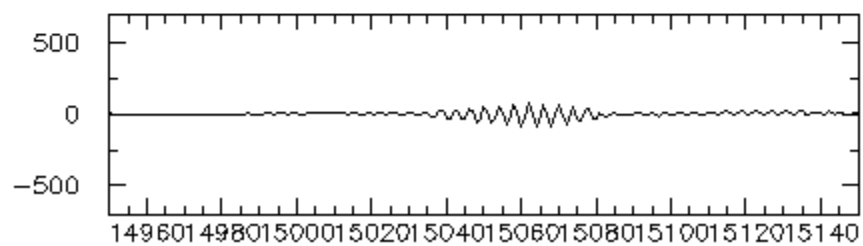
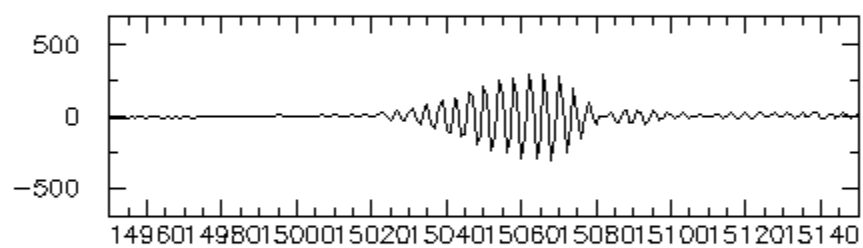
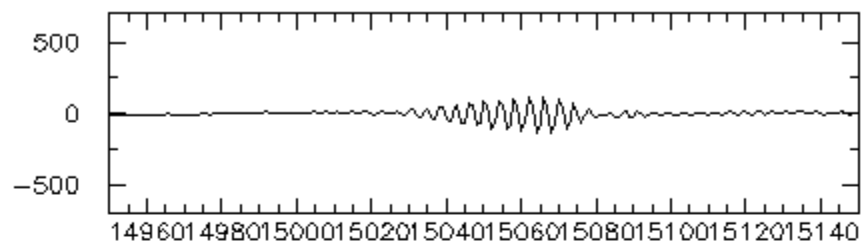


graph8.png 16:05

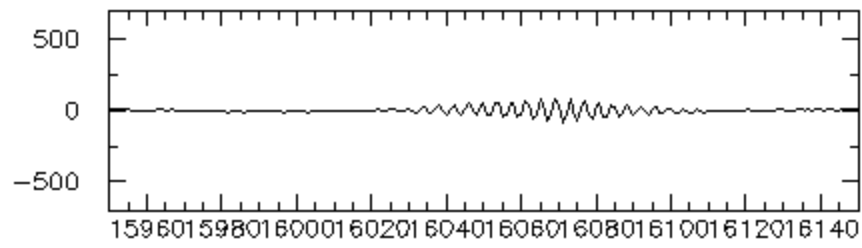
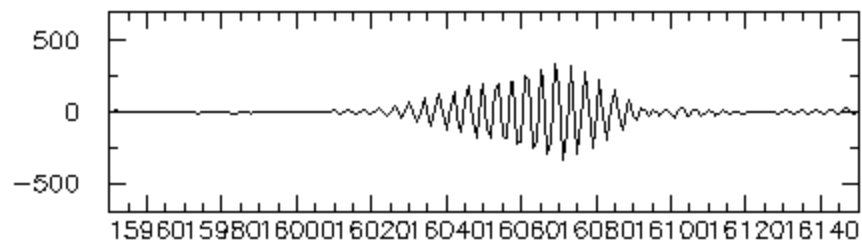
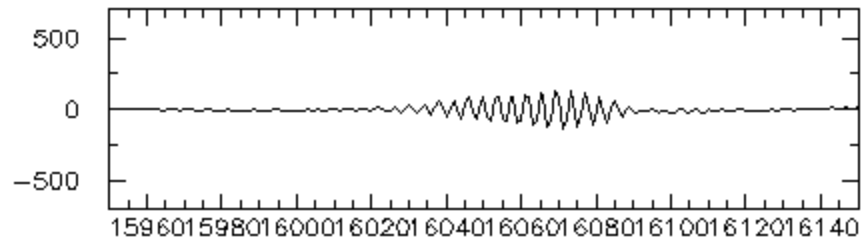
graph9.png shows same thing happening 1000 turns later, starting at turn 14000.
graph10.png starts at turn 15000. graph11.png starts at 16000. graph12 at 17000.
graph13 at 18000. graph14 is 19000. graph15 is 20000.



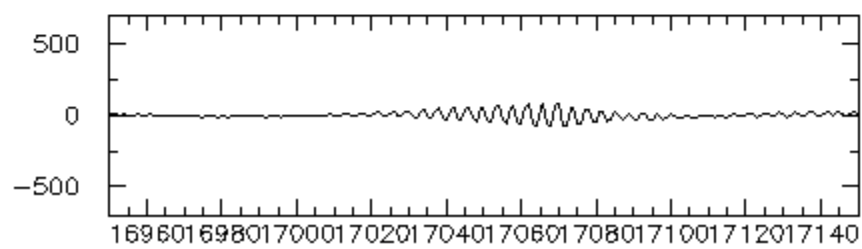
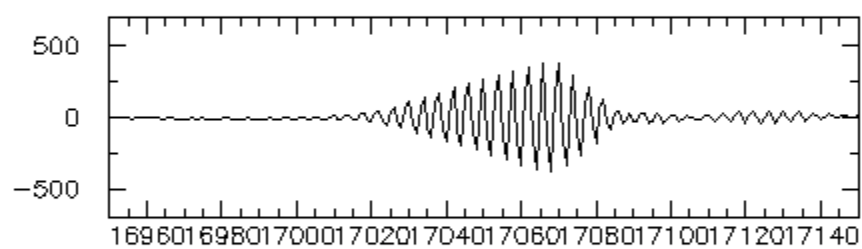
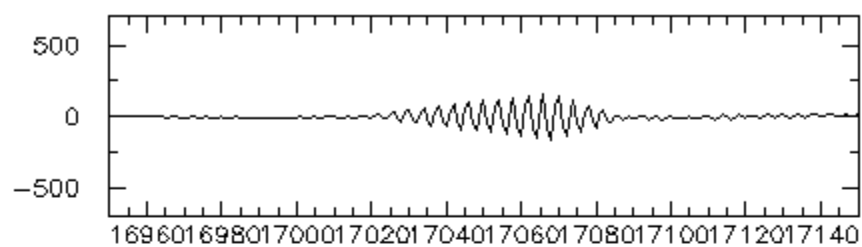
graph9 16:09



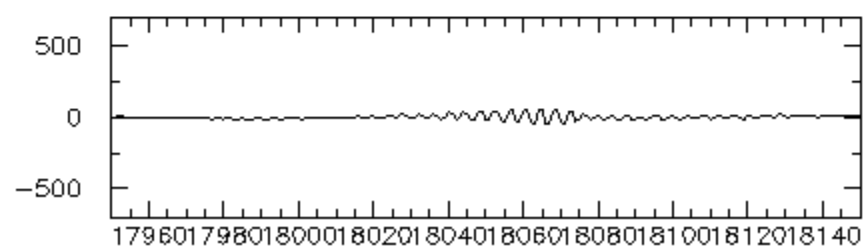
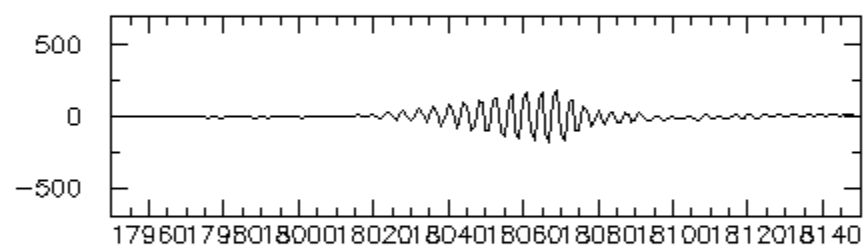
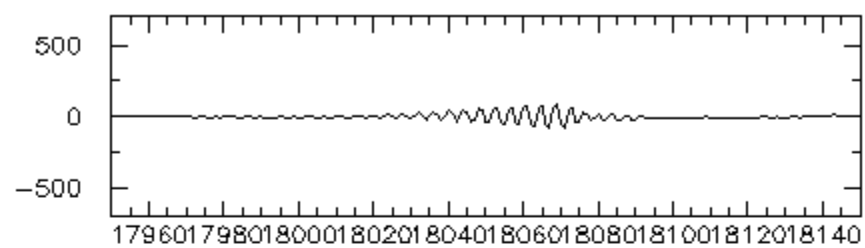
graph10 16:13



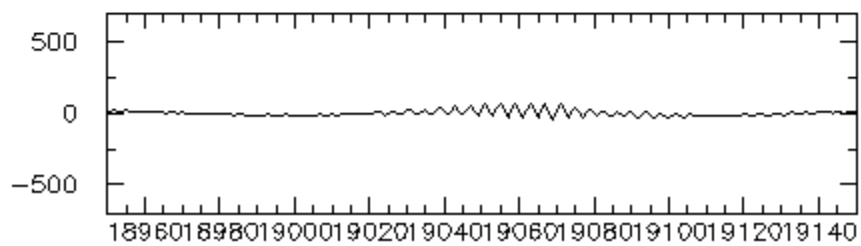
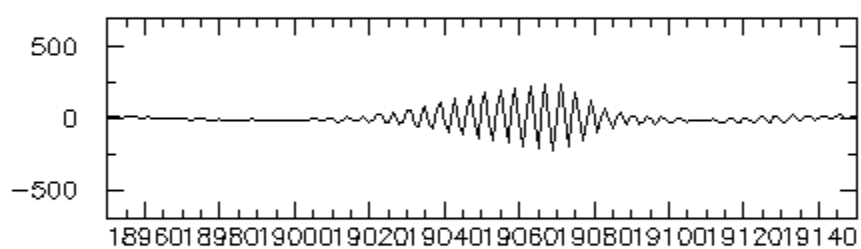
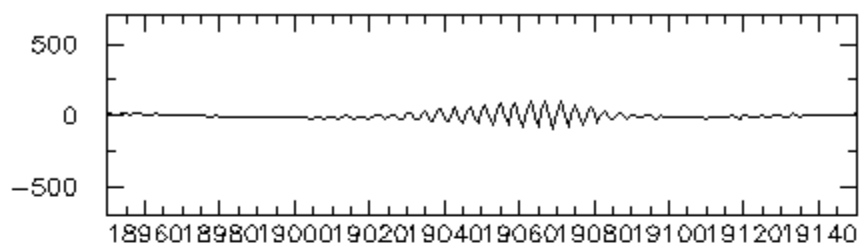
graph11 16:16



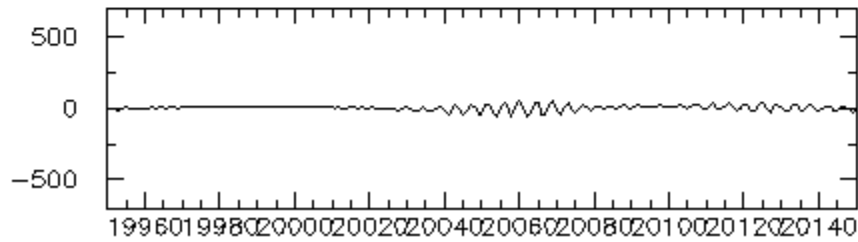
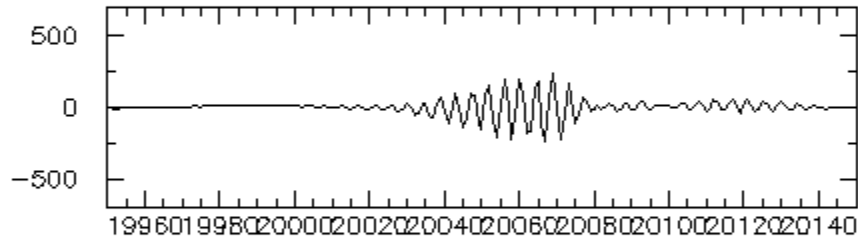
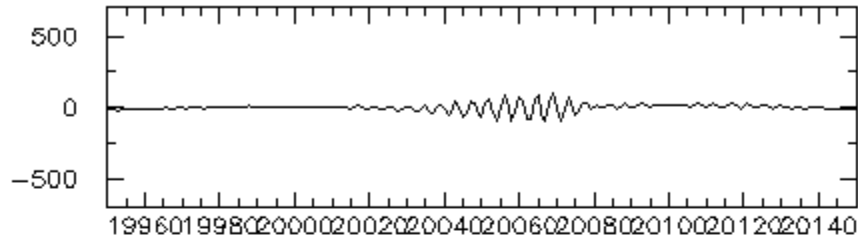
graph12 16:18



graph13 16:20

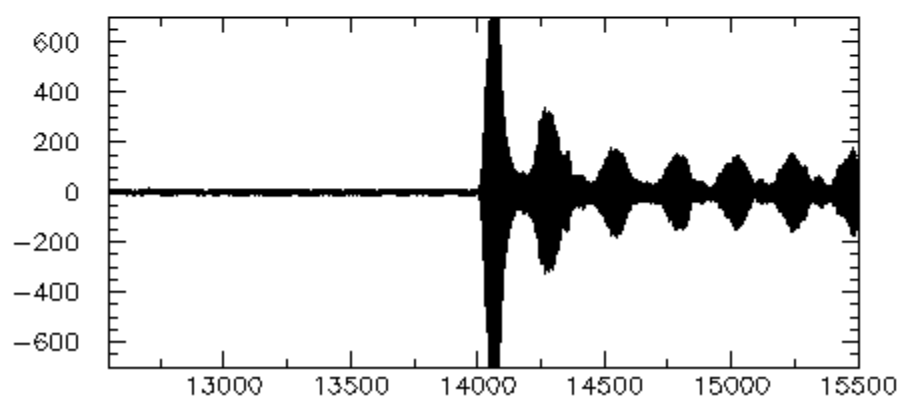
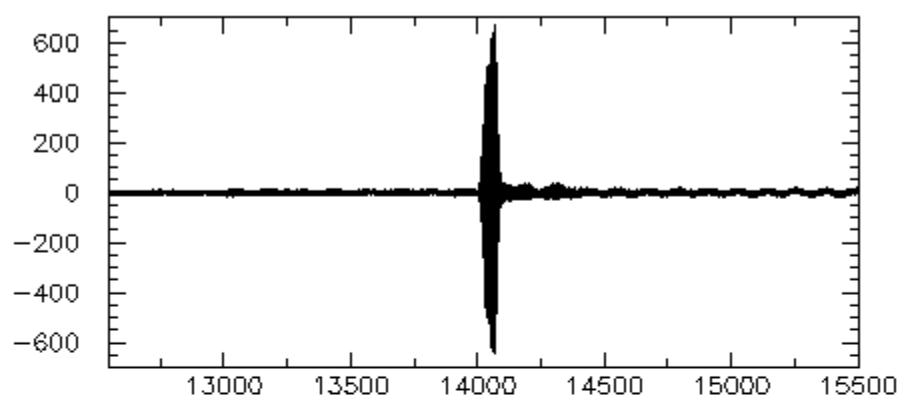


graph14 16:26

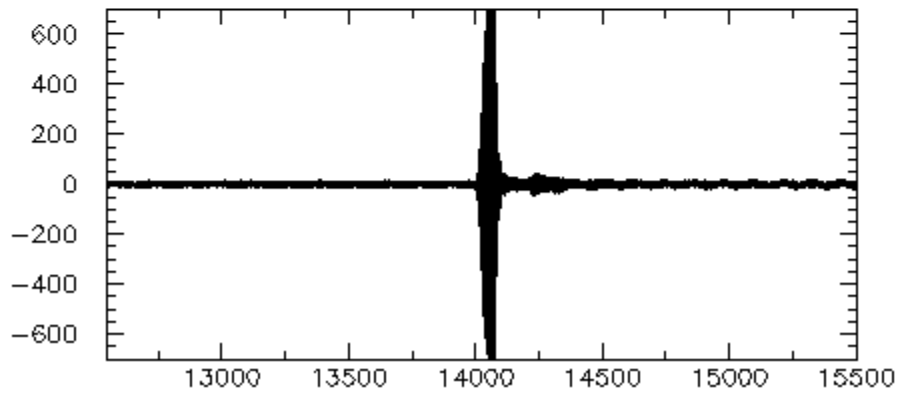
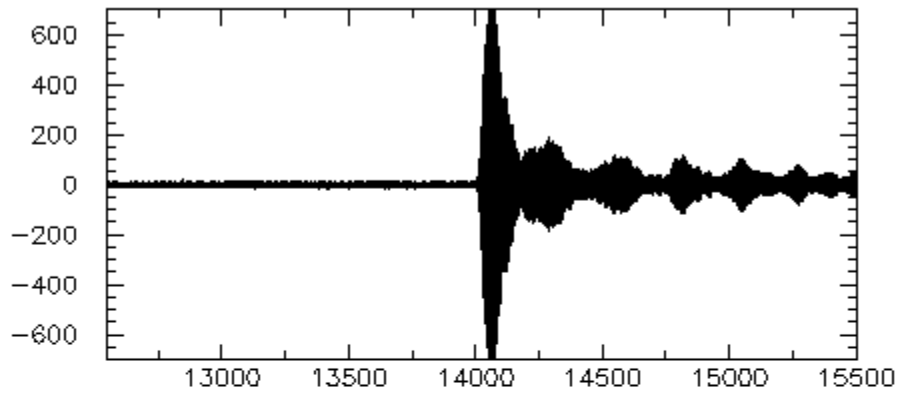


graph15 16:32

graph16 shows qq-qq2 (i.e. quadrature minus two-turn-delayed quadrature) for bunches 18-59 (readout notation), with damping active for bunches 20-59, antidamping for 70 turns and then (top) damping or (bottom) turning off the damper. Good stuff!! graph17 is same thing for all 84 bunches!

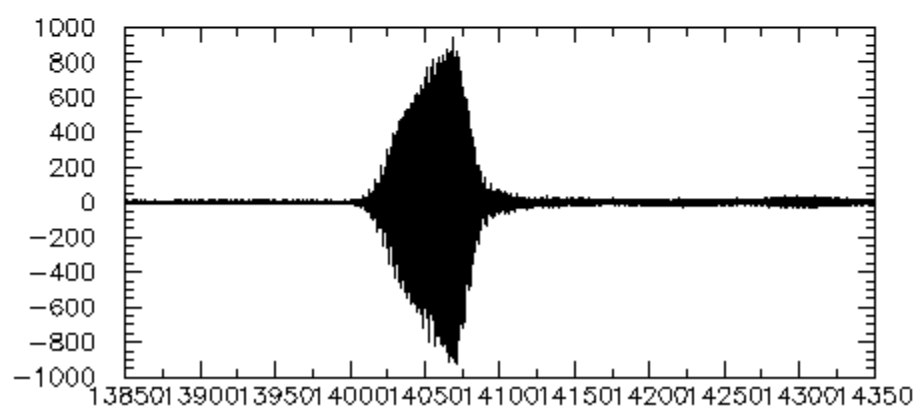
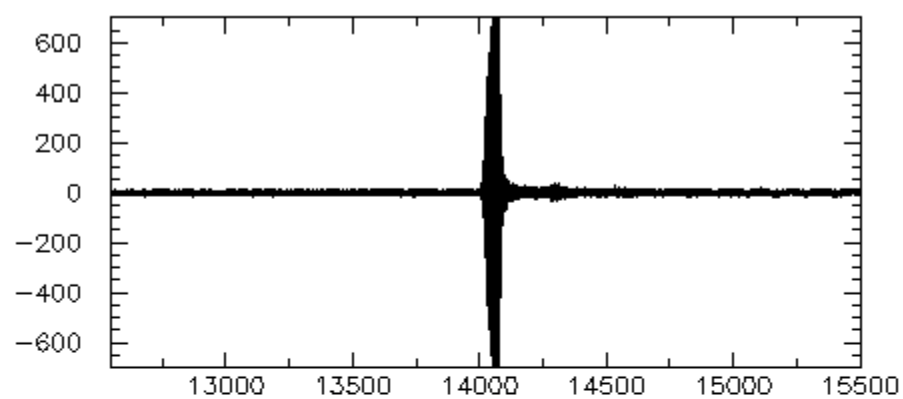


graph16 16:43

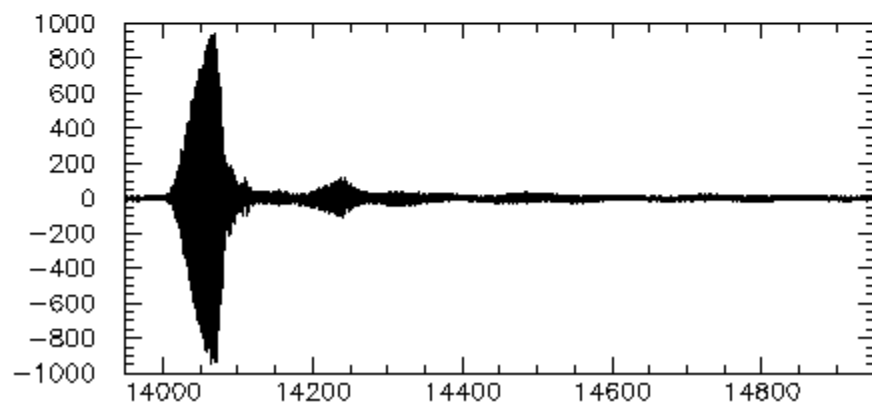
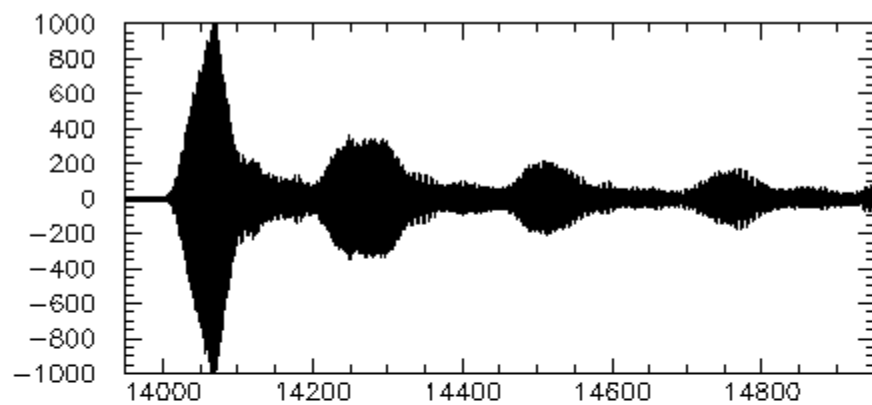


graph17 17:00

graph18 is after loading the program that kicks only 6/8 of each bucket. lower plot is zoomed in horizontally and zoomed out a bit vertically.

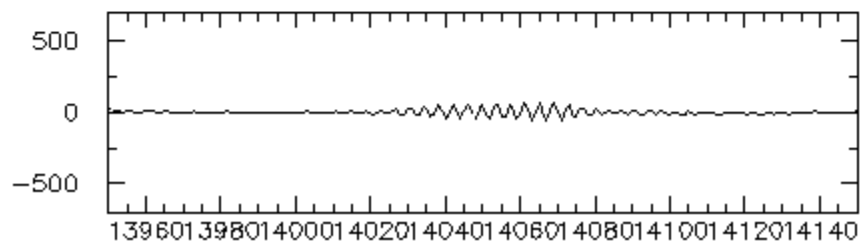
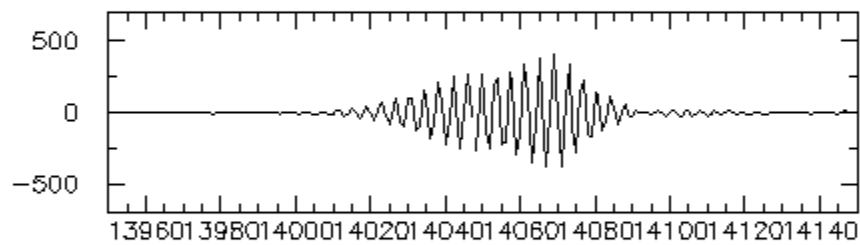
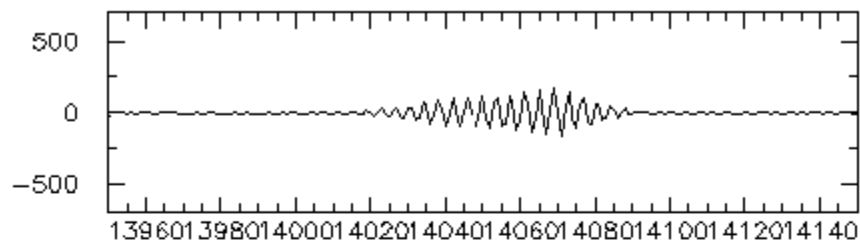


graph18 17:17



graph19 18:20

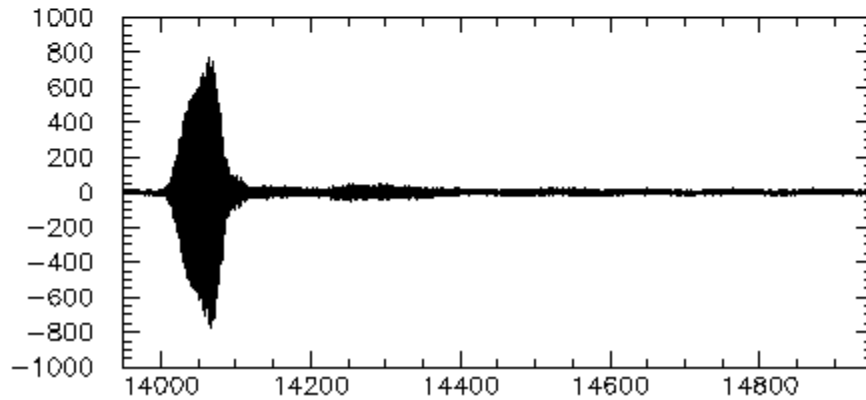
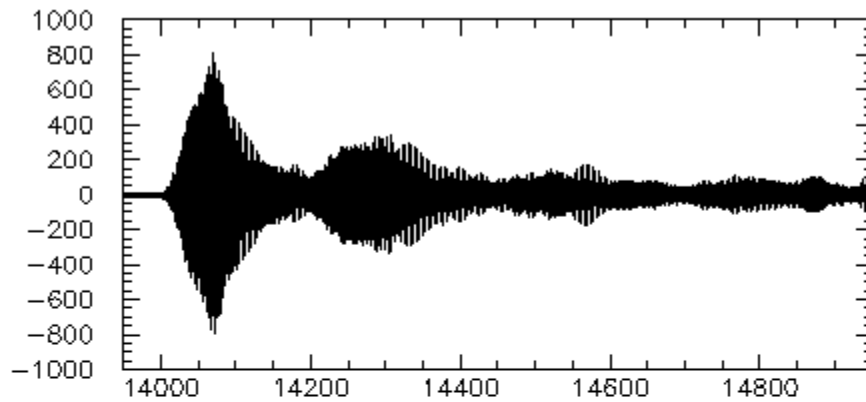
Here's a repeat of graph9 after the recompile mentioned above, for shorter kicks. It doesn't seem to make a difference.



graph20 18:28

I added the X damper readout logic (but not yet filter or kick logic) on ADC channel 0 and recompiled. Compile time went up from 8 to 10 minutes -- not too bad. I am rerunning the last antidamp/damp test on the Y damper, to make sure I didn't break it. Looks good!

graph21.png.



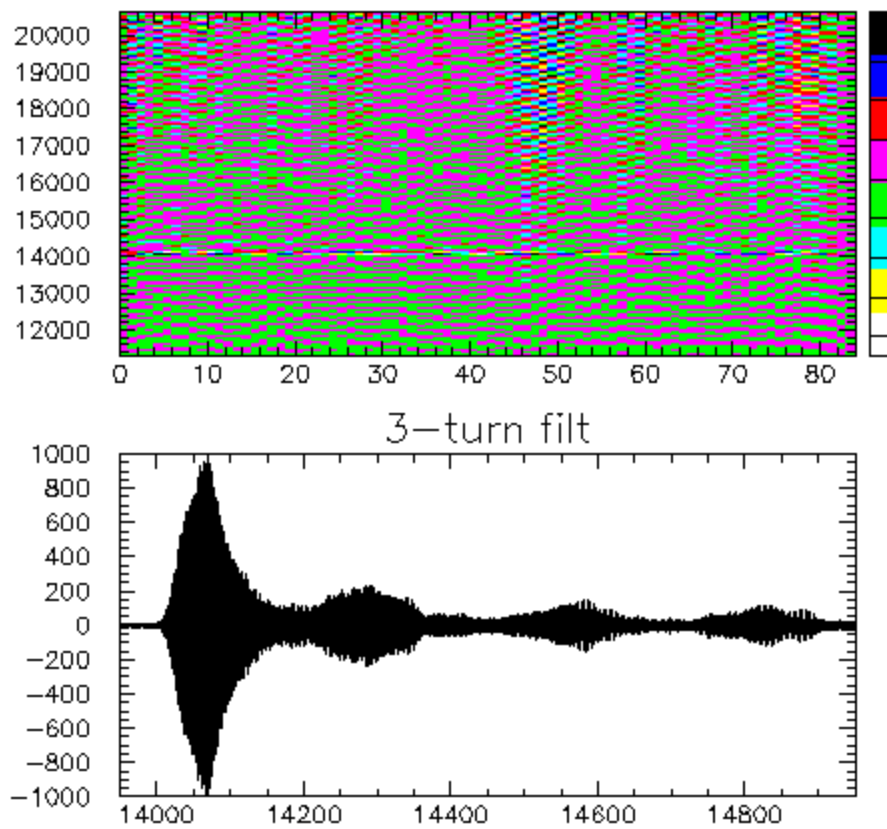
graph21 19:17

Well, I tried to get the X position read out bunch-by-bunch and turn-by-turn, so that I could look to see whether the oscillatory feature in e.g. the top half of graph21 is (as suspected) coupling between Y and X planes. I made progress in that direction but didn't quite get there. It's time for dinner. Not a bad day!

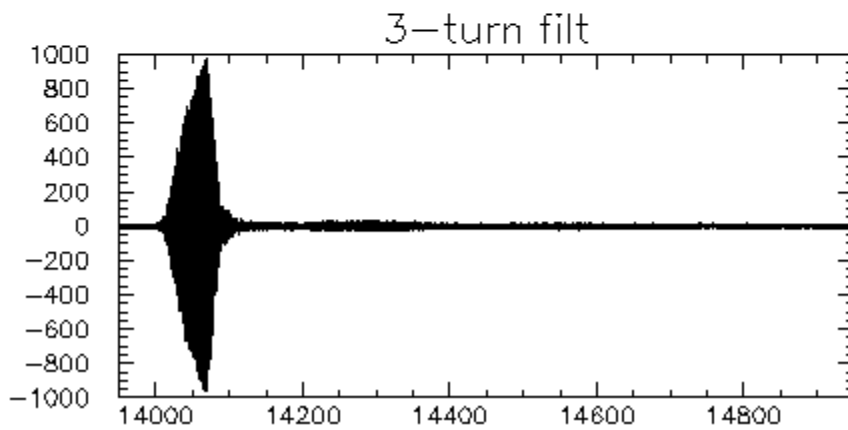
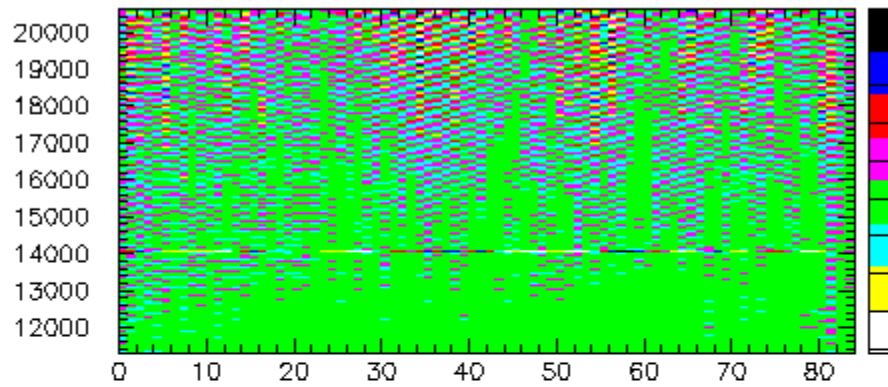
2008-07-10

Nathan and I did some rearranging. I want to make sure I didn't break anything. We took data on one \$17 pulse and saw signals, so at least I didn't mess up the pickup or RF clock cables. Now let's try damping. Amplifiers on. It looks as if I did at least break the oscilloscope hook-up. graph22.png (15:46) shows antidamping without then damping. OK! graph23.png (15:51) shows antidamping then damping. Works! This is with amplifier gain at 100%. Saved file as fifo_amp100pct_nathan_001.paw. ballpark exponential

damping time is 12 turns.

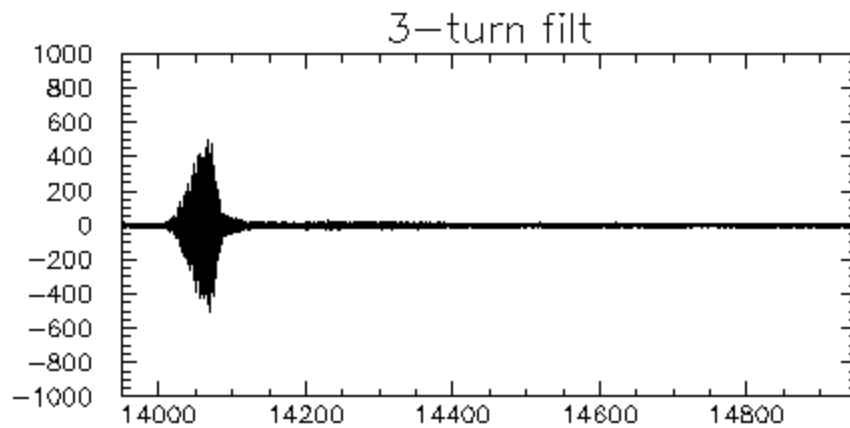


graph22.png



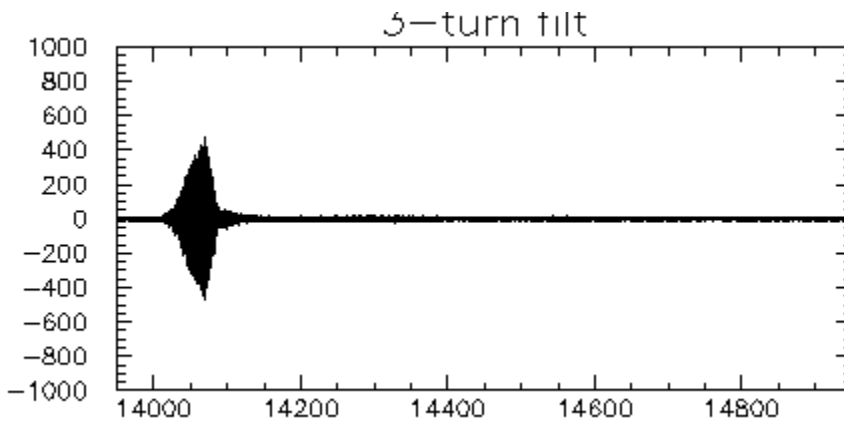
graph23.png

Now let's turn off one plate's amplifier and see what happens. graph24. still damps! Looks like same damping time, but amplitude after 70 turns of antidamping was 40% of full-power case. fifo_ampl100pct1plate_nathan_002.paw.



graph24.png

Both plates on at 20% gain. Did nothing. Then changed D4YBBG from 7 to 31, and saw graph25 (16:26).



graph25.png

Note to self. We need to cook up a simple interactive fit for (anti)damping rate, either linear or exponential.

We conclude that driving just one plate works, so we can use one amplifier per plane for now. But we also found a third amplifier, so most likely we will use just one amplifier for the vertical plane and begin working on the horizontal plane with two amplifiers.

Most of the day was consumed by a big cleanup effort over in the damper area.